CONTRIBUTIONS TO RECOGNITION AND LOW BIT RATE CODING OF SPEECH

A thesis presented for the degree of
Doctor of Philosophy
in Electrical & Electronic Engineering
at the
University of Canterbury,
Christchurch, New Zealand.

This thesis has been
accepted for the degree of
Master of Engineering in
Electrical & Electronic
Engineering

by
Lim Ching Aun
B. E. (Hons 1), M. E.
March 1992

Abstract

Original research contributions to aspects of automatic recognition and low bit rate coding of speech are presented. A comprehensive review of the historical development of speech processing is provided. Various facets of the physiology of the human speech production organs, the classifications of speech sounds, and human perception of speech sounds are discussed.

Essential mathematical theories for speech processing are outlined. Three different techniques for automatic speech recognition are discussed in detail. One of these techniques is implemented, modified and extended. It is found that these modifications and extensions result in improved recognition rate of the digits zero to nine uttered by several New Zealand speakers. These results are comparable to those of studies conducted elsewhere.

A novel speech coding scheme is examined. It is found that the scheme produces good quality speech at 16 kbps. However, there are some unpleasant click sounds which are believed to be caused by the instability of the scheme. The scheme is modified by adding an impulse function to the glottal pulse. This is found to be effective in stabilising the scheme and in removing the click sounds.

Because errors are inevitable in any speech coding scheme, it is important to quantify these errors and to analyse their nature, as the analysis may provide useful hints on ways in which these errors can be eliminated. To this end, the reconstructed speech is assessed alongside two reference signals, which are formed by adding known amounts of multiplicative and additive noise, respectively, to the original speech. These assessments strongly suggest that the errors in the reconstructed speech are largely caused by the multiplicative noise.

Acknowledgements

The real friend, someone has said, should always appear to us to be our strongest enemy. He is the one who tells us our faults when no one else cares about us enough to risk the telling. Our truest friend openly, though not judgingly, challenges what is not right about us and therefore he threatens our self-esteem. Yet we know that he does so for our own good, that he cares about us. Therefore, however much we may fail to appreciate or even resent what he says or does, deep down we are thankful for his real friendship. The same also applies, I think, to the real master. I am referring to, of course, my supervisor since the final year of my undergraduate degree, the late Professor Richard H.T. Bates. Not only does he make sure that I work hard, but he also leads by his own example. Not only does he criticise, he also takes criticism. His constant encouragement and confidence in me have played an important part in the completion of this thesis. In return, I can only hope to repay his mentorship by doing for others what he has done for me.

I am most thankful to members of the staff (both academic and technical) especially Mr Bill Kennedy, Senior Lecturer in the Department of Electrical & Electronic Engineering, University of Canterbury, Christchurch, New Zealand for many practical suggestions.

The following members of the speech group deserve special thanks: Dr C. William Thorpe for his tutelage in the fine art of the Shift-And-Add (SAA) and the CLEAN techniques, which he has combined together to form a novel speech coding technique, and which I have subsequently modified. The results of the modifications are detailed in Chapter 6. Mr Andrew Elder for the use of the vector quantization algorithm which he implemented and Ms Tracy Clark for sharing the speech database. I am appreciative of the support rendered by other members of the speech group.

I have enjoyed very much the company and the 'wits' of Dr Alan Murch, Dr Ross Murch, Mr Abbas Safa, Mr Tony Enright, Mr Charles Parker, Mr Quek Bek Kim, Mr Andrew Ross and Mr Vaughn Smith, with all of whom I have shared the same study room R7 over the last few years.

I have lived in College House, a hall of residence at the University of Canterbury, throughout most of my university time in New Zealand and there are many people associated with College House that I want to extend my thanks to: the present Principal, Mr Tony Brough, the ex-Principal, Rev. Canon Robert G. McCullough

and my tutor colleagues for many interesting after dinner discussions; the cleaning and kitchen staff for their unrelenting kindness and care in looking after my basic needs of food and shelter; the office staff and the overseas students for many hours of fun and a lot of other things, *e.g. friendship*, that money cannot buy.

The help of my flatmate, Mr Cheah Wei Chun in preparing some of the illustrations in this thesis is appreciated. I am greatful for the assistance rendered by Mr Tan Wai Ming in the evaluation of the modified speech coding scheme.

I am grateful to my family for their support and I acknowledge the award of a postgraduate scholarship from the University Grants Committee of New Zealand [1].

Finally, I would like to say that writing this thesis has been an *intensely personal* experience. There have been moments of jubilation (when the all too infrequent splash of an idea has occurred) *and* great frustrations. However, all of these, I am sure, are shared by all the RHTBOBs (Note: B = B + G and thanks Philippa!) scattered around the corners of the earth! This knowledge has given me enormous strength in completing this thesis.

[1] Now the Vice-Chancellor Committee of New Zealand

Contents

Preface

Speech research began in the Electrical & Electronic Engineering Department of this university in the late 1970s. It grew out of studies of computer-controlled aids for musicians (Tucker *et al.*, 1977) and methods for extracting pitch information from speech and musical sounds (Tucker and Bates, 1978; Brieseman, 1984). That research led to the investigation of micro-computer based speech aids for disabled people. The incorporation of digital signal processor (DSP) integrated circuits into these speech aids has enhanced their potential usefulness because of their real-time capabilities (Turner, 1986b).

With the financial support of the Telethon Trust for the Year of the Disabled, the University Grants Committee and the Unisys Linc Development Centre, the present speech group now consists of eight postgraduate students and three academic staff. Current research includes the development of micro-computer based tools for speech therapists, techniques of low data rate speech encoding and the development of algorithms for reliable word and speaker recognition (Bates *et al.*, 1987). As well as employing standard speech analysis techniques, this research has benefited from the novel (to speech processing) techniques introduced by Professor Bates from his wide experience in other areas of information processing.

I joined the speech group on the completion of my Master of Engineering studies in March 1988, and I started investigating two aspects of speech processing research. The first involves a review of isolated word recognition algorithms and the implementation of one of the algorithms. My motivations for engaging in this has been realising that, with the use of computer technology becoming more widespread, the impetus towards simplifying the communication with and through computer-based machines has steadily increased. Because speech is the easiest and the most natural human communication medium, it would be highly convenient and profitable if one could communicate with these machines via speech rather than via a keyboard. It is obvious that an automatic speech recognition system which can recognise and translate natural speech into computer commands, text or data has many applications such as aids for the disabled, dictaphones for secretaries, booking and information services, toys, computer mail services by voice, *et cetera*. Therefore, the benefits and rewards to be associated with the successful development of such a system are so great that the goal is worth pursuing. My research is related to this goal.

The second aspect of speech processing studied and extended in this thesis is a locally developed scheme to digitally encode speech. The underlying goal of speech encoding schemes is to transmit speech with the highest possible quality, over the channel with the least possible capacity, and at the smallest possible cost. With the

almost unlimited channel capacity promised by optical fibres, the need for speech encoding schemes which efficiently utilize the channel capacity seems unjustifiable at first glance. On closer examination, however, it becomes clear that there are many situations such as satellite communication links, cellular phone systems and microwave links where efficient use of channel capacity is an important consideration.

The cost of speech encoding typically increases with coder complexity which, in turn, often means higher coder efficiency and channel utilization. The spectre of high cost has, in the past, deterred studies of complex (and potentially efficient) digital speech encoders. However, advances in large scale integration technologies have dramatically reduced the cost of integrated devices. This has triggered a renewed interest in the search for novel and sophisticated coding schemes for speech.

This thesis is written in seven chapters. Each chapter concludes with a summary of the main points raised in that chapter. The contents of this thesis are summarised in the following paragraphs, which also identify my original research contributions.

Chapter 1 explains the detailed motivation for conducting the research reported in this thesis. Then follows an introduction to the physiology of the human speech production organs and the linguistic aspects and characteristics of speech. This leads to the source-filter model of speech production. The physiology of the human ear and various aspects of human hearing are also discussed.

Chapter 2 reviews the historical development of the four main areas of speech processing: namely speech synthesis, speech analysis, speech recognition and speech coding. From these reviews, one can gain an understanding of the interplay between these four areas and how development in one area affects the development in the others.

Chapter 3 lays down the mathematical background of established techniques employed in speech processing research. In particular, the linear predictive and vector quantization techniques are described in detail.

Chapter 4 introduces the first part of my research and reviews three of the most commonly used strategies for speech recognition. The basics of the dynamic time warping, hidden Markov models and neural networks are presented. Several important issues regarding the implementation of these approaches are discussed, followed by a comparison of the relative merits of each.

In Chapter 5, I detail the original work that I have carried out in speech recognition using the hiddem Markov model (HMM). I have written all the software implementing the HMM. This software was written to interface to **sigproc**, which is a signal processing package developed, princicipally by Nigel Brieseman (formerly a postgraduate student in this department) and extended by various postgraduate students in the department. The experiments that I have carried out are discussed. The results of these experiments are analysed.

Chapter 6 introduces the second part of my original research, *i.e.* low bit rate speech coding. I outline a new method for a low bit rate speech coding scheme based on Shift-And-Add (SAA) and "CLEAN". SAA/CLEAN was first developed and demonstrated as a viable coding scheme by Thorpe (1990). While the scheme produces intelligibible speech at 16 kbps, the reconstructed speech contains unpleas-

ant "click" sounds. At the suggestion of my supervisor, the late Prof R.H.T. Bates, I have made some modifications to the SAA/CLEAN coding scheme in an attempt to remove these "click" sounds. The modifications are described in this Chapter. The performance of the modified SAA/CLEAN coding scheme at 16, 8 and 4 kbps is assessed using the Mean Opinion Score (MOS) tests. The results of these assessments are analysed.

Finally, Chapter 7 draws conclusions on the research reported herein and offers suggestions for further investigations on the areas studied.

Papers and conference presentation that I have authored or co-authored but have not been presented for examination for any degree are listed below:

LIM, C.A., ELDER, A.G., CLARK, T.M. and BATES, R.H.T. (1990), 'Software Implementation of hidden Markov model for recognition of isolated digits uttered by New Zealand speaker', In *Proc. NELCON*, Auckand, September, pp. 287–294.

LIM, C.A. (1991), 'Modification and evaluation of a speech coding scheme', In *Proc. NELCON*, Palmerston North, August, pp. 67–74

P.H. Gardenier, C.A. Lim, D.G.H. Tan and R.H.T. Bates,(1986), 'Aperture distribution phase from single radiation pattern measurement via Gerchberg-Saxton algorithm', In *Electronics Letters*, Vol. 22 No. 2, pp113-115.

P.H. Gardenier, C.A. Lim, D.G.H. Tan and R.H.T. Bates,(1986), 'Feed-position and reflector-shape errors of satellite communications antenna from radiation pattern magnitude', Presented at IPENZ Conference 86, Auckland University, New Zealand, February 1986.

P.H. Gardenier, C.A. Lim and C.R. Parker,(1988), 'Satellite communications antenna misalignments inferred from far field magnitude', Proceedings of the 25th New Zealand National Electronics Conference", NELCON, Christchurch, August 1988, pp.83-88.

The results embodied in this thesis have also been presented at numerous weekly research seminars within the Department of Electrical & Electronic Engineering, University of Canterbury, Christchurch, New Zealand and demonstrated on several occasions to the representatives from the Unisys Linc Development Centre, Christchurch and MITI, Japan.

Chapter 1

INTRODUCTION

"In the beginning was the word . . . "

(St. John 1.1)

This thesis is concerned with 1) the implementation of an isolated word recognition algorithm and 2) the study of a new scheme for encoding speech at low bit rates.

This chapter begins by outlining the motivation for conducting the research reported herein. In order to help the reader to understand the later chapters, the physiology of the human speech production organs and the classes of human speech are reviewed. A mathematical model of the human speech production mechanism is then formulated. This is followed by a discussion on the human hearing organism. The chapter concludes with a discussion on various aspects of the perception of human speech.

1.1 Motivation for speech recognition research

The computer revolution has given us new power to process information in almost every facet of our daily life. Effective techniques of human-machine interactions are necessary if this power is to be harnessed by computer-illiterate people. As we become more and more dependent on this powerful tool in our day-to-day existence, it becomes increasingly more annoying that one has to communicate with it via a keyboard and not via human speech.

Speech is our everyday, informal, communication medium. It helps human beings to become acquainted and it allows human minds to interact. Emotions and thoughts can be expressed and records of knowledge can be transmitted and preserved. We

are unique among life forms in our ability to acquire and use speech. It is even more remarkable that we have derived the ability from physiological apparatus designed for other purposes - the vital functions of breathing and eating.

There are two aspects of human-machine communication via speech: the outputing of computer responses as speech and the inputing of computer commands using speech. The first involves the synthesis of speech while the second involves machine recognition of speech. In many ways, speech synthesis is technically much simpler than machine recognition of speech because a human being can adapt to a machine's way of speaking much more easily than the machine could adapt to the human's.

One good reason for communicating with machines via speech, instead of spending the time typing away at the keyboard, or reading the computer print out, is that our hands and eyes would then be free for other tasks. For those without keyboard skills, it would also be highly convenient to have an automatic speech recognition system which can recognise and translate natural speech into computer commands, text or data. Moreover, speech is emitted virtually omnidirectionally, and does not require a free line of sight. Related to this is the use of speech as a secondary medium for status reports and warning messages. Occasional interruptions by voice need not interfere with other activities, unless they demand unusual concentration. People can assimilate spoken messages and queue them for later action quite easily and naturally.

Moreover, unless machines interact with ordinary people in ways in which they feel at ease, computers will remain the province of technically trained people and continue to be treated with suspicion by everyone else. Once such interaction is achieved, the enormous potential of computers to aid and enhance human life will be made available to all.

Another important feature of speech communication stems from the universality of the telephone receiver itself and the existence of a world-wide distribution network. One can go into a phone booth anywhere in the world, carrying no special equipment, and have access to anybody's phone within seconds. If speech input to machines were routine, this would mean instant access to one's computer from virtually anywhere. This gives speech a substantial advantage over information transfer by the telephone network using a modem and a visual display unit. The data input problem is far from solved as yet, despite the availability of limited word recognisers (Wallich, 1987), or means of inputting via the touchtone telephone keypad (or a portable calculator-sized tone generator). Easy remote access without special equipment would thus be a great and unique asset to speech communication.

A further advantage of interacting with computers via speech is that it is potentially very cheap. Being all-electronic, except for their loudspeakers, speech systems are well suited to high-volume, low-cost, low scale integration (LSI) manufacture (Witten, 1982). Other computer output devices are at present tied either to mechanical moving parts or to the cathode ray tube (CRT). This was realised quickly by the computer hobbies market, where speech output peripherals have been selling very well since the mid 1970s.

A further point in favour of speech is that it is natural-seeming and somehow cuddlier when compared with printers or visual display units (VDU). There are many

more advantages to communicating with computers via speech. For example, speech

- can be used in the dark
- can be varied from a (confidential) whisper to a (loud) shout
- requires very little energy
- is not appreciably affected by weightlessness or vibration

Useful as it is at present, speech output would be even more attractive if it could be coupled with speech input. Many of the benefits of speech output are even more striking for speech input. Although human beings can assimilate information faster through the eyes than the ears, the majority can generate information faster with the mouth than with the hands. Rapid typing is a relatively uncommon skill, and even high typing rates are much slower than speaking rates (although whether one can originate ideas quickly enough to keep up with fast speech is another matter!). To take full advantage of the telephone for interaction with machines, machine recognition of speech is obviously necessary. A microwave oven, calculator, pin-ball machine, or alarm clock that responds to spoken commands is certainly more attractive than one that just generates spoken status messages.

All of this points to the need for achieving recognition of speech by computers. While it is desirable to have a versatile speaker-independent recognition system that is capable of recognising casual speech, it will be some time (centuries maybe?) before this goal is reached (Bates *et al.*, 1988). A less ambitious task, but still far from being attained, is the automatic recognition of conversational (or professional or technical) English uttered under controlled conditions by any person prepared to speak carefully and clearly (Rabiner and Levinson, 1985). Although there has been some success for this approach in highly specialised applications (see Table 1.1), there is such variability between individual speakers that it seems to make much better sense at present to concentrate on the problem of speaker-dependent speech recognition. The aim of this research is to develop a system that will recognise isolated words uttered by New Zealand speakers. The benefits and rewards to be associated with the successful development of such a system, even if only of the speaker-dependent, isolated word vocabulary type reported in this thesis, are so great that the goal is obviously worth pursuing.

1.2 Motivation for speech coding research

The other major part of this thesis is the study of a new scheme for encoding speech at low bit rate. As mentioned earlier, human beings communicate most naturally and efficiently via speech. This coupled with the widespread availability of telephone services has increased the use of speech as a communication medium and provided the impetus for the emergence of many new digital coding techniques that enhance the applicability of voice communications and storage. These techniques allow more speech to be represented with a given number of binary digits, without losing natural voice quality.

REFERENCE	VOCABULARY	TRAINING MODE	ACCURACY (%)
Itakura (1975)	A-Z, 0-9	one speaker	88.6
Rosenberg and Itakura (1976)	84 words	ten speakers	91.6
Rabiner *et al.* (1979)	A-Z, 0-9, STOP, ERROR, REPEAT	28 speakers	79
Hermansky (1987)	isolated digits	48 males/females	91
Austin and Fallside (1988)	connected digits	one speakers	90
Furui (1988)	1011 words	20 males	98
Huang and Jack (1988)	isolated digits	2 males 2 females	95.8 - 98.8
Drews *et al.* (1989)	up to 1000 words	speaker-independent	unspecified
Fissore *et al.* (1989)	1011 words	5 males 2 females	94.6
Lee *et al.* (1990)	997 words	speaker-independent	96

Table 1.1: *Summary of several word recognition systems.*

The advanced coding techniques just becoming available will yield natural-sounding telephone speech at digital transmission rates of 16, 8 and eventually 4 kilobits per second, not just the 64 and 32 kbps that have become international standards (Jayant, 1986).

Although high-bandwidth channels and networks are becoming more viable, coding speech at low bit rates has retained its importance. One reason is the growing need to transmit speech messages with a high level of security over low data-rate channels, such as radio links. Another factor is the desire for memory-efficient systems for voice storage, voice response, and "voice-mail."

Low bit rate coding of voice is critical for accommodating more users on channels that have inherent limitations of bandwidth or power - like cellular radio or satellite links. It can lend flexibility to the design of the evolving integrated-services digital network (ISDN), which will reduce communication signals - voice, graphics, video, or computer data - to the common denominator of binary digit sequences. In particular, low bit-rate voice coding can ease the transition to shared channels for voice and data. Further, the low bit rates can readily adapt voice messages for packet switching and help to make voice mail practical and popular. Encryption of sensitive messages can become more readily available to business as well as to the military, and the capacity of recoding devices like answering machine can rise dramatically.

The new low bit rate speech coding scheme expounded in this thesis is based on the deconvolution technique "CLEAN" which (like shift-and-add) arose in the astronomical field (Högbom, 1974) and has also been applied in other areas of signal processing (Bates, 1981b). The technique was first applied for low bit rate speech coding by Thorpe (1990). The goal is to achieve acceptable (to the consumer) quality speech at a data rate of about 16 kbits/sec (and eventually at 8 kbits/sec) and to assess its performance against other coding schemes like multi-pulse LPC (Kroon, 1986; Singhal and Atal, 1989).

1.3 Physiology of the human speech production organs

Speaking is initiated at the brain which formulates the messages, thoughts and feelings that are to be expressed. Appropriate lexicographic and grammatical rules are then invoked, and the words are coded into a sequence of neuromuscular signals which control the various parts of the speech production organs as shown in Figure 1.1.

The first physiological process involved in speaking is called inhalation. This involves expanding the rib cage and lowering the diaphragm, thus creating a drop in pressure in the lungs. The drop in pressure then draws air into the lungs. This process is then reversed by contracting the rib cage and raising the diaphragm, thus increasing the pressure in the lungs. The increased pressure forces the air to flow up the trachea or wind pipe.

At the top of the trachea, the air encounters the larynx which is a bony structure to which the vocal cords are attached. Figure 1.2 shows a cut-away view of the larynx and the vocal cords. Initially, the muscles in the larynx are tensed, and hence the cords are pulled together, trapping air behind the vocal cords. Then, air

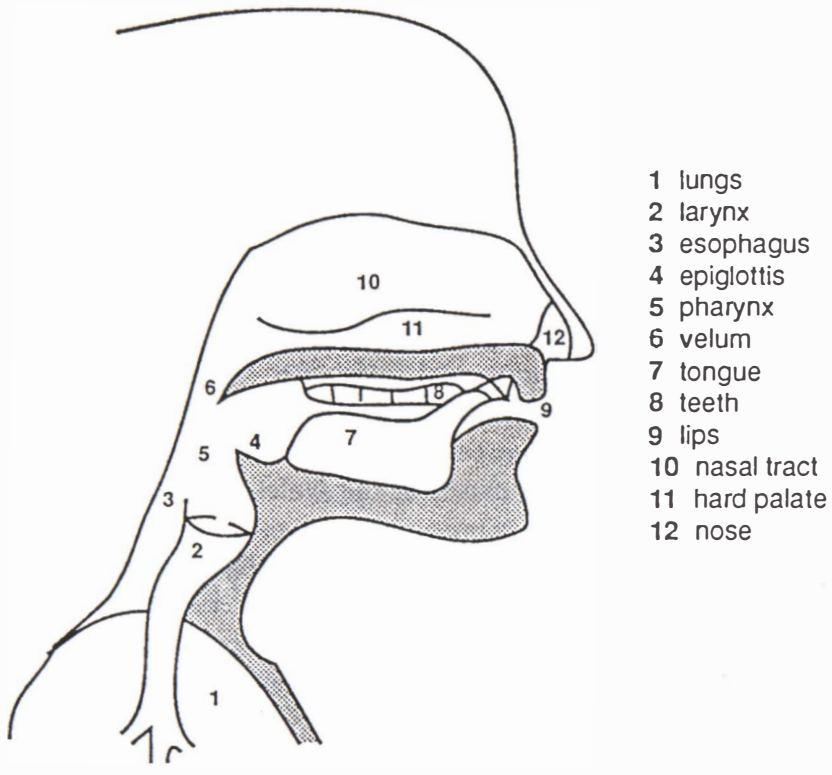


Figure 1.1: *The human speech production organs.*

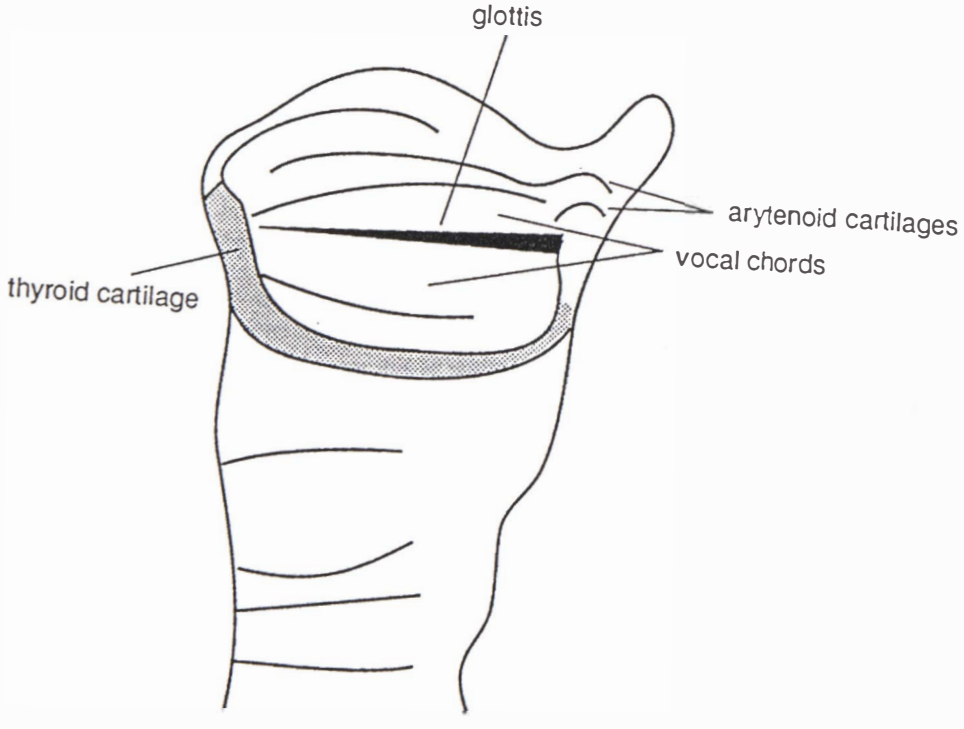


Figure 1.2: *The human larynx and vocal cords.*

Figure 1.3: *The time-pressure waveform of a typical glottal pulse train. This lasts for 10ms for a typical male and 5ms for a typical female.*

pressure gradually builds up below the vocal cords until it is sufficient to force the cords apart, allowing the trapped air to escape through the slit-like opening between the vocal cords known as the glottis (see Figure 1.2). The flow of air through the glottis causes a local drop in pressure, a phenomenon known as the Bernoulli effect (Ainsworth, 1988). This drop in pressure allows the tension in the laryngeal muscles to close the glottis, hence interrupting the flow of air. The pressure then builds up again, forcing the vocal cords apart, and allowing the air flow to continue. In this manner, the cycle repeats itself, producing a quasi-periodic pulses of air emanating from the glottis. By varying the tensions of the cords and the air pressure from the lungs, humans can control the frequency of these vocal cord vibrations over a typical range of 50-500 Hz for adults, and up to 1000 Hz for children. At this point, it should be noted that frequency of vibration of the vocal cords is almost entirely determined by the mass and the tension of the cords themselves, and is relatively independent of the resonant frequencies of the vocal tract (Linggard, 1985). This source/system separation is fundamental to the validity of the source-filter model of speech production.

A typical glottal pulse waveform is shown in Figure 1.3. The shape of glottal pulse waveform is usually quite simple and may be approximated by a triangular pulse (Witten, 1982), or shift-and-add pulse (Brieseman *et al.*, 1987). The spectrum of a train of glottal pulse waveform tends to be a series of harmonics decaying at about 12 dB/octave (Bates *et al.*, 1988).

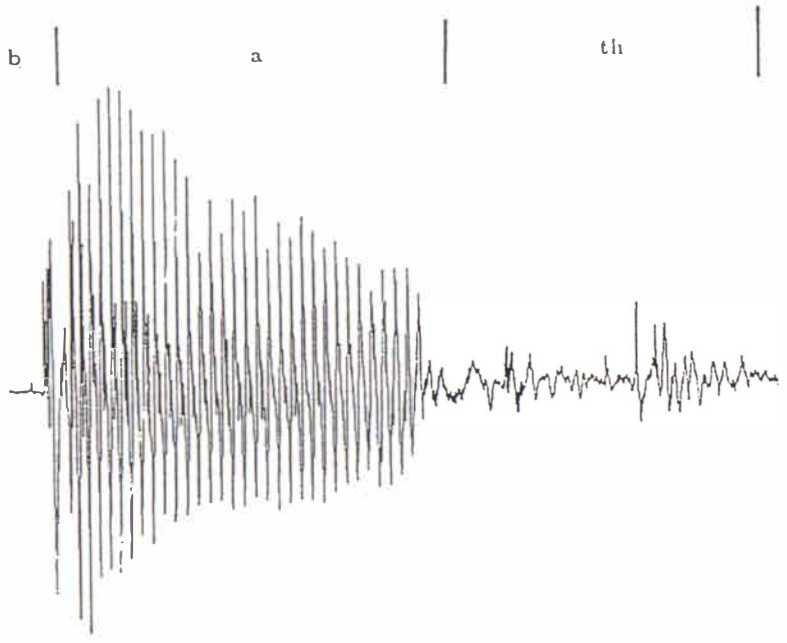


Figure 1.4: *The time-pressure waveform for the word "bath".*

Beyond the glottis is a branched air-filled channel some 17 cm long (see Figure 1.1) known as the vocal tract. The first section of the vocal tract is the pharynx. The pharynx is a tube which extends from the esophagus (manifested externally, more prominently in males, as the "Adam's apple") and the larynx to the base of the skull. The pharynx is composed primarily of constrictor muscles and is quite flexible. Above the pharynx, the tract splits into the oral and nasal cavities, which are separated by the velum (or soft palate) and the bony hard palate. The velum is a muscular flap that, when lowered, couples the nasal cavity to the rest of the vocal tract. The oral cavity (containing the tongue, teeth and lips) extends from the epiglottis, which covers the larynx during swallowing.

The vocal tract acts as a resonant cavity for the glottal pulse waveform. It amplifies the harmonics of the glottal waveform which lie near the natural resonances of the tract while attenuates the others. By changing the shape of the vocal tract, we can control the resultant acoustic pressure wave (and hence the sounds) that radiates into the air. Figure 1.4 shows the pressure waveform emitted by a New Zealand speaker uttering the word "bath".

The shape of the vocal tract are changed by altering the positions of the tongue, the lips, the jaw and the velum. When air has passed from the glottis through the pharynx, it may be diverted at the velum, partly or wholly, into the nasal cavity and out through the nostrils. This gives rise to a class of sounds known as 'nasal'. Two examples of nasal sounds are the /m/ and /n/ sounds. The degree of nasality is set by how much the velum is lowered or raised to allow air to flow through the cavity.

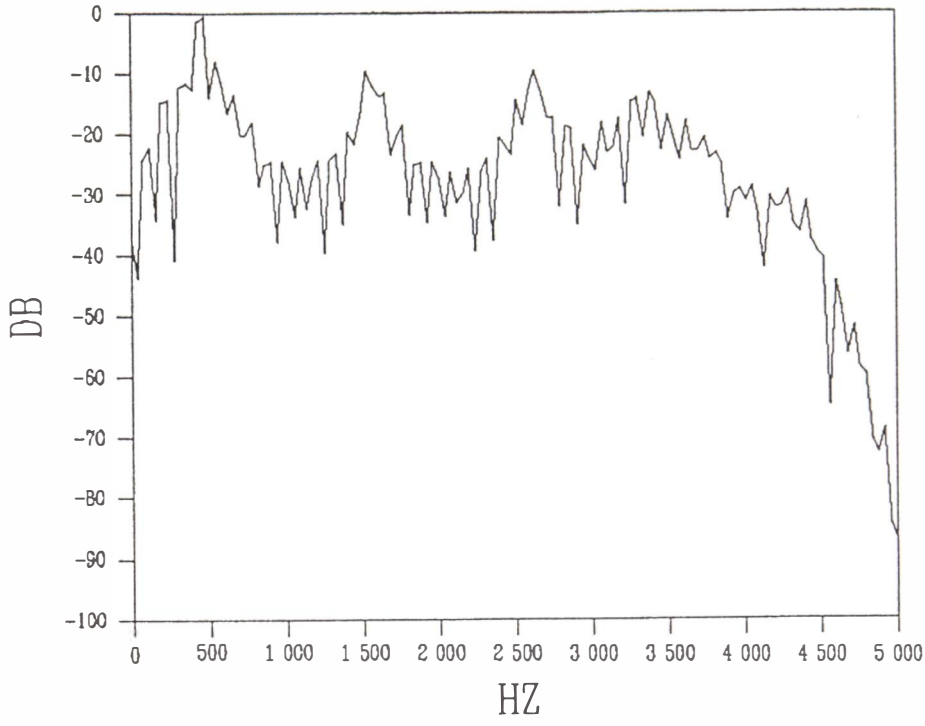


Figure 1.5: *Spectrum of voiced speech. Notice the fine-periodic structure of 100 Hz which corresponds to the frequency response of the glottal pulse train and the formant structures.*

Unlike the nasal cavity where the shape is fixed, the oral cavity can assume a wide range of shape. The jaw can be opened or closed to a varying degree, and to a lesser extent it can be moved forward or backward. Similarly, the lips can be opened or closed, pushed-out or drawn-back, and used in opposition to the tongue and teeth. The most flexible part is the tongue. As well as moving forward or backward and raising or lowering, it can also assume a variety of shapes. It can be a flat plate or a round lump, the sides can be folded up to form a half tube, and the tip of the tongue can be curled back or pushed forward between the teeth.

1.4 Classification of speech sounds

Speech sounds can be classified into three main types: *voiced, unvoiced* and *plosives* according to the ways in which they are generated.

Voiced speech is generated by a quasi-periodic glottal pulse train exciting the vocal tract which acts as a resonant cavity. The production of quasi-periodic pulse train has already been described in §1.3. The vowels are examples of voiced speech. Voiced speech is easily identified in the time domain pressure waveform (see Figure 1.4). This is because the waveforms of voiced speech are quasi-periodic and are of higher amplitudes. The part of the waveform containing the vowel /a/ is indicated in Figure 1.4.

Figure 1.5 shows the spectrum of a segment of voiced speech. From Figure 1.5,

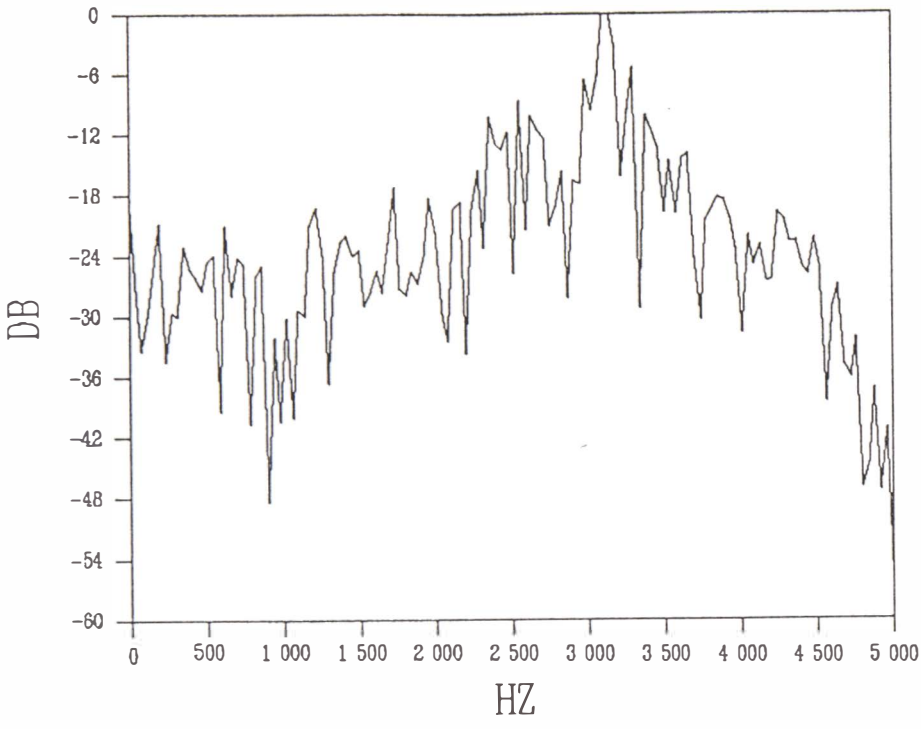


Figure 1.6: *Typical spectrum of unvoiced speech.*

one can see a fine-periodic structure of about 100 Hz superimposed on a slow-varying envelope. The fine-periodic structure and the envelope correspond to the frequency responses of the glottal pulse train and the vocal tract filter respectively. Figure 1.5 also shows the formants (or resonant frequencies) of voiced speech. A significant point is that, although most of the energy in voiced speech is concentrated below 1 kHz, the portion of the spectrum above that limit contains most of the speech information (Bates *et al.*, 1988).

Unvoiced speech sounds are generated when air is forced through a narrow constriction formed within the vocal tract. For example, during whisper, the vocal cords are held slightly apart so that a turbulence is created when air passes through them. The turbulence causes a noisy excitation of the resonant cavity and *aspirated* sounds are created. Such sounds occur in the *h* of "hello", and for a very short time after the lips are opened at the beginning of "pit" (Witten, 1982).

Other unvoiced sounds such as *ss, sh,* and *f* are produced by constrictions made in the mouth. For example, to produce the *ss* sound, the tip of the tongue is moved high up, very close to the roof of the mouth. Turbulent air passing through this constriction causes a random noise excitation known as "frication". In *sh* sound, the tongue is flattend close to the roof of the mouth slightly further back, in a position somewhat similar to *ee*, but with a narrower constriction; while *f* is produced with the upper teeth and lower lip.

A typical spectrum of unvoiced speech is shown in Figure 1.6. Comparing Figure 1.6 with Figure 1.5, it can be seen that for unvoiced speech, most of the

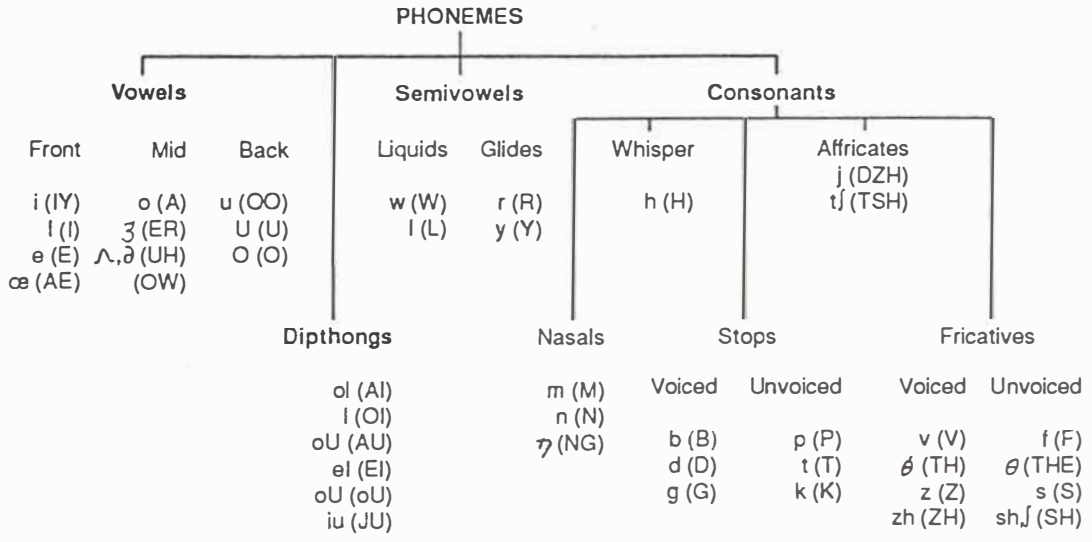


Figure 1.7: *The phonemes of the English language.*

energies tend to be above 2.5 kHz.

The third class of speech sounds is the plosive sounds. In order to produce plosive sounds, the vocal tract is completely closed at some point, so the flow of air stops, allowing the pressure to build up behind a complete closure. Then the pressure is suddenly released, giving rise to impulsive excitation. Thus, the time waveform of a plosive is characterised by a silent period (identifiable as the parts with low amplitudes samples), followed by a sudden burst of high amplitudes samples. The plosive /b/ is illustrated in Figure 1.4.

Plosives may be voiced or unvoiced. With voiced plosives, the vocal cords are allowed to vibrate at, or before the instant of the release of pressure (Ainsworth, 1988). Examples of voiced plosives are /b/, /d/ and /g/ sounds. A /b/ is produced by closing the vocal tract with the lips, /d/ with the tip of the tongue on the tooth ridge while /g/ is produced by closure at the velum. On the other hand, the unvoiced plosives are produced in a similar manner to their voiced counterparts, except that the vocal cords are not allowed to vibrate until some 50 ms after the instant of release. Examples of unvoiced plosives are /p/, /t/ and /k/.

1.5 Phonetic transcription of speech sounds

A *phoneme* is defined as a set of distinctive sounds from which any words in the language under consideration can be built up. Most experts in linguistics consider English to comprise the 42 phonemes as shown in Figure 1.7. As indicated

in Figure 1.7, the phonemes themselves can be classified into four groups : vowels, dipthongs, semivowels and consonants.

The shape of the vocal tract is maintained more or less fixed during a vowel, which is of course a voiced sound. A dipthong is also a voiced sound, but it involves a gliding transition of the vocal tract between a pair of shapes characterising two phonemes. One of the phonemes is dominant because of its greater duration, while the second phoneme, being shorter, is called the glide. A semivowel , such as "W", "L", "R" and "Y", is similar to a vowel in that it is a voiced sound. It is not, however, a continuous sound in the same sense as a vowel. The other major class of phonemes are the consonants, which can be divided up into four further classes: stops, fricatives, affricatives and nasals. Stops occur when there is a complete closure of air flow along the vocal tract as in "B" and "P". They can be voiced or unvoiced. A more continuous consonant is the fricative which is formed by forcing air through a constriction, for example "V","F" and "S". Affricatives are a combination of a stop and a fricative as in "ch" in "chair". Finally, nasals ("M","N","NG") are produced by creating resonances in the nasal cavity (Turner, 1986a).

1.6 Source-filter model of speech production

From the discussions in §1.3 on the production of human speech, one can conclude that any segment of speech, $s(t)$ can be expressed as

$$s(t) = e(t) \odot v(t) \odot l(t) \tag{1.1}$$

where $\odot$ denotes the convolution operation, t represents time while $e(t), v(t)$ and $l(t)$ denote the excitation source, impulse response of the vocal tract and that of the lips respectively. Implicit in Equation (1.1) is that the speech segment under consideration is short enough so that the vocal tract changes negligibly throughout its duration. This condition is satisfied if t is of the order of 10 - 20 ms (Bates *et al.*, 1988). By lumping together the responses of the vocal tract and the lips, Equation (1.1) can be rewritten as

$$s(t) = e(t) \odot h(t) \tag{1.2}$$

Equation (1.2) is the well-known source-filter model developed by Fant (1960). Another important assumption is that the source and filter are considered to be independent of each other. A block diagram of the source-filter model, as expressed by Equation (1.2), is illustrated in Figure 1.8.

A more useful representation of the source-filter model can be obtained by performing a Z transform operation (Oppenheim and Schafer, 1975) on Equation (1.2). Thus, using the notations $S(z) \longleftrightarrow s(t)$, $E(z) \longleftrightarrow e(t)$ and $H(z) \longleftrightarrow h(t)$ to mean that $S(z)$ is the Z transform of $s(t)$, *et cetera*, Equation (1.2) becomes

$$S(z) = E(z)H(z) \tag{1.3}$$

by the convolution theorem. In Equation (1.3), $H(z)$ is usually considered to be an all pole filter (Bates *et al.*, 1988), so it is convenient to write

$$H(z) = \frac{1}{1 - A(z)} \tag{1.4}$$

and

$$S(z) = \frac{E(z)}{1 - A(z)} \tag{1.5}$$

where $A(z)$ is a polynomial function of z

$$A(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{1.6}$$

One can re-arrange Equation (1.5) as a prediction equation, where the current sample of speech $s(n)$ is predicted from the previous p speech samples and the excitation signal, *i.e.*

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n) \tag{1.7}$$

In speech signals recorded from human speakers, the values of the a_k and $e(n)$ are unknown *a priori*, but they can be estimated using a least squares approach (Makhoul, 1975). The a_ks vary slowly as different sounds are uttered, but can be considered constant for short intervals (of the order of 10-20msec). If the estimates of a_ks are available, then the prediction error $e(n)$ is the difference between the real speech sample and the predicted speech sample, *i.e.*

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{1.8}$$

The mean square error over N samples is then

$$E(N) = \sum_{n=1}^{N} [s(n) - \sum_{k=1}^{p} s(n-k)a_k]^2 \tag{1.9}$$

At this point, it suffices to state several methods can be invoked to solve for an optimum set of a_ks such that $E(N)$ is minimised. The a_ks are known as the linear predictive coefficients (LPC). Further details are provided in Chapter 3.

1.7 Physiology of the ear

The primary organ involved in hearing is the ear. Figure 1.9 shows a simplified section through the ear. As shown in Figure 1.9, the ear is divided into three main parts, the external, middle and inner ears (Flanagan, 1972a).

The external ear consists of the *pinna* which surrounds the entrance to the external ear canal. The *tympanic membrane*, commonly known as the ear drum, marks the boundary between the external and the middle ear. The ear canal channels sound pressure wave to the ear drum, causing it to vibrate. The canal or *meatus* can be approximated by a uniform pipe which is open at one end and close at the other. Therefore, it has normal mode of vibration which occurs at frequencies where the pipe length is an odd multiple of a quarter wavelength. Since the speed of waves travelling in air at $20°c$ is about $344m/s$ and the average length of the canal is about $2.7cm$, this means that the first mode of vibration occurs at about $3000Hz$

$(= \frac{344}{4(0.027)})$. This resonance helps the ears' sensitivity in this frequency range. Measurements by Wiener and Ross (1946) have shown that the sound pressure at the ear drum is in fact about $5 - 10dB$ over the value at the canal entrance.

In the middle ear, the sound waves that cause small movements of the ear drum are transmitted by the lever action of the *ossicles* to the oval window (see Figure 1.9). The ossicles consist of three small bones, which because of their shapes, are called the hammer (*malleus*), the anvil (*incus*), and the stirrup (*stapes*). As a result of the lever action of these bones, the force exerted on the oval window is about three times that exerted on the ear drum. Further, because the area of the oval window is much smaller than that of the ear drum, the pressure on the former is $18 - 20$ times higher than that on the latter (Hills and Shipley, 1985).

The ear has two ingenious mechanisms to protect the delicate inner ear against very intense sounds. Firstly, the stapes vibrates in a different mode in response to intense sounds, so that less energy is transferred to the inner ear. Secondly, two small muscles draw the stapes away from the oval window and the ear drum inwards by reflex action, thus reducing the amount of energy transferred to the inner ear (Flanagan, 1972a).

The inner ear, or *cochlea*, consists of a coiled, snail-like tube filled with liquid. Figure 1.10 shows a simplified diagram of an uncoiled cochlea. As shown in Figure 1.10, the *basilar membrane* stretches longitudinally along the axis of the cochlea, dividing it into two cavities. The basilar membrane is filled with some 30,000 hair cells on which the endings of the auditory nerve terminate. The vibration of the stapes causes the oval window to vibrate. The movement of the oval window in turn induces pressure waves in the fluid inside the cochlea. As the resulting pressure waves propagate along the upper cavity of the cochlea, the basil membrane is displaced. The mechanical motion of the basilar membrane is then converted into neural pulses which are transmitted to the brain by the auditory nerve. These neural pulses are then interpreted to give the sensation of sound (Flanagan, 1972a).

1.8 Human perception of speech sounds

One way of approaching the problem of speech recognition is to study the perception of speech by human. This is a reasonable approach because the only completely adequate speech recogniser we have at present is the human auditory system and the brain.

In this section, the different types of human *perceptual* responses to sounds is discussed. In particular, the perception of *loudness* and *pitch* are dealt with in §1.8.1 and §1.8.2 respectively. In addition, the concepts of *masking effects* and *critical bands* are treated in §1.8.3.

1.8.1 Loudness

Loudness reflects our *perceptual* response to the amplitude of a speech signal. For example, when we say that one sound is "louder" than another, we are usually referring to the relative amplitudes of the pressure waves of the two sounds. However,

while amplitude, *i.e.* the air pressure of the speech signal, is the primary acoustic "correlate" of the percept of loudness, psychoacoustics tests have found that human judgement of loudness is a function of the duration as well as the amplitude of the sound, especially when the sound lasts for less than one half of a second (Lifschitz, 1933). Further, it has been established that the perception of loudness is also a function of frequency (Thorpe, 1990). Thus, two pure tones that are of the same amplitude but of different frequency may not be perceived as equally loud (see Figure 1.11.)

An objective measure of loudness is the *sound pressure level*, expressed in decibel (dB). The decibel is defined with respect to a fixed air pressure reference. It is equal to

$$20 \log_{10} \frac{P}{2 \times 10^{-5}} \tag{1.10}$$

where P is the measured pressure in N/m^2. The reference pressure of $2 \times 10^{-5} N/m^{-2}$ is chosen because it is the threshold of hearing (Lieberman and Blumstein, 1988). Figure 1.12 shows the sound pressure levels of some typical sounds and noises.

The perceptual unit of loudness is the *phon*. The phon scale is calibrated against the decibel scale using a 1000-Hz pure tone as the reference frequency. With this calibration, the phon scale is numerically the same as the decibel scale. For example, the loudness of a 1000-Hz pure tone with a sound pressure level of 5 decibels is 5 phons. Figure 1.13 shows the conversion between the decibel scale and the phon scale and the typical range of loudness encountered in our daily life.

1.8.2 Pitch

The fundamental frequency of a sound is normally regarded as its pitch. However, psychoacoustics tests (Lieberman and Blumstein, 1988; Fant, 1973) have shown that this definition of pitch does not agree entirely with human perception of pitch. For example, if a pure tone of 1000 Hz is played to a human subject and the human subject is then asked to adjust the control of a frequency generator until a sound which has twice the pitch of the 1000 Hz signal, the subject will not set the control to 2000 Hz. Instead, the subject will select a frequency of about 3000 Hz (Lieberman and Blumstein, 1988). Hence, the ratio of the perceived pitch of two sounds is not linearly related to the ratio of their fundamental frequencies.

A linear scale for pitch is the mel scale. In this scale, the pitch of a pure tone with a mel value of 1000 is perceived to be twice as high as one with a mel value of 500. The mel scale can be related to the fundamental frequency by the use of a conversion graph as shown in Figure 1.14. Another interesting characteristic of pitch perception is that the fundamental frequency need not be present (Houtsma and Goldstein, 1972) in order to be perceived by a listener.

1.8.3 Masking and critical bands

Masking is the obscuring of one sound by another. Much research has been carried out to study the nature of masking and its mechanisms (Jeffress, 1970). Early work by Mayer (1894) found that it is easier to mask out a tone using another tone of lower

frequency than one of higher frequency. The experiments by Wegel and Lane (1924) supported Mayer's theory. In addition, Wegel and Lane (1924) discovered that the masking effect is more pronounced if the frequency of the second tone is near (< 10 Hz higher or lower than) that of the first tone. This is attributed to the occurrence of "beating" when two sounds of slightly different frequencies are received simultaneously by both ears. It is interesting to note that beating vanishes when the tones are presented simultaneously, but separately to opposite ears (Thorpe, 1990).

Fletcher (1940) proposed the concept of critical bands to explain the masking phenomenon. Fletcher studied the effect of masking by white noise and found that only a certain bandwidth centred around the tone to be masked actually contributed to masking. He called this bandwidth that contributed to the masking effect the critical bandwidth. Subsequent results by Zwicker (1961) and Sharf (1970) have shown that the human audible frequency range (20 Hz - 20 kHz) can be divided into 24 critical bands as shown in Table 1.2. From Table 1.2, it can be seen that the bandwidth of each band is narrower at the lower frequencies than at the higher frequencies. The existence of these critical bands strongly suggest that the cochlea acts like a bank of overlapping band pass filters (Thorpe, 1990).

Greenwood (1961) suggested that the critical bands represented equal distances (about 1.3 mm) along the basilar membrane of the ear. This was consistent with the findings of (Bekesy, 1960) who observed where the maximum displacements of the basilar membrane occur, in response to tones of different frequencies. Later, Zwislocki (1965) proposed that these critical bands may correspond to neural density in the cochlea.

1.8.4 Theories of perception

From the many and varied psychoacoustics experiments (Thorpe, 1990; Ainsworth, 1988; Lieberman and Blumstein, 1988) that have been carried out to study human perception, two theories of perception have emerged to account for the results of these experiments.

The first formulation of the theory of perception was proposed by Halle and Stevens (1959). In this theory, it is believed that the listener perceived speech by "recognising" the articulatory gesture and the whole neural mechanisms that are used to control the entire speech production process and "hypothetically produced" the sounds. This was named analysis by synthesis by Halle and Stevens (1959). This was extended by Liberman *et al.* (1967) who introduced the motor theory of speech perception to explain various aspects of speech perception, including the perception of intonation and stress. More recent formulations of the motor theory of speech perception (Liberman and Mattingly, 1985) claims that there is a one-to-one mapping between a linguistic construct and a motor command. However, these claims have been found to be inconsistent with the experimental data that has been derived over the past fifty years (Lieberman and Blumstein, 1988).

The second theory of how humans perceived speech sounds is called the speech mode of perception. Thus, when subjected to speech sounds and other acoustic signals such as music, human listeners behave in a different "mode". More electrical

Number	Centre frequencies	Cut-off frequencies	Bandwidth (Hz)
1	50	100	80
2	150	200	100
3	250	300	100
4	350	400	100
5	450	510	110
6	570	630	120
7	700	770	140
8	840	920	150
9	1000	1080	160
10	1170	1270	190
11	1370	1480	210
12	1600	1720	240
13	1850	2000	280
14	2150	2320	320
15	2500	2700	380
16	2900	3150	450
17	3400	3700	550
18	4000	4400	700
19	4800	5300	900
20	5800	6400	1100
21	7000	7700	1300
22	8500	9500	1800
23	10500	12000	2500
24	13500	15500	3500

Table 1.2: *Critical bands of human hearing. After Zwicker (1961).*

activities are recorded in the left hemisphere (for right-handed person) when speech sounds are played (Lieberman and Blumstein, 1988). However, no observable differences in electrical activity are recorded in the two hemispheres of the brain when listeners are subjected to musical sounds (McAdam and Whitaker, 1971).

1.9 Human capacity for differentiation of sounds

It is very useful to have an idea of human capability in differentiating various types of sounds. Many experiments (Pollack, 1952; Stevens and Davis, 1938; Hawley (ed), 1977) have been conducted to determine the limits of this capacity. These experiments generally falls into two categories: discrimination versus identification. These two different types of differentiation represents quite different perceptual tasks.

The example which follows will make the distinction between these two ways of differentiating types of sounds clear. Imagine your friend wants to test your ability to differentiate sounds make by a guitar when a note is struck. You are of course blind-folded. Your friend play two notes, one following the other. Your friend then ask you to identify whether the two sounds are the same or different. Provided that the guitar is in good repair and tune and your hearing is normal, you should be able to discriminate between all the notes of a guitar. The results of an *identification* tests will be quite different. This time, a note is played on the guitar and you are asked to identify the note of each key that is struck in a random order. This is difficult even if you are one of the few people with perfect pitch. Psychoacoustics experiments have established that most people can reliably identify no more than about four or five different tones (Pollack, 1952). On the other hand, if you have normal hearing and can hear the frequency range between 20Hz and 20kHz, you can discriminate about 350 000 different sinusoidal tones (Stevens and Davis, 1938).

Kimura (1961) conducted an experiment to test the human capability for recognition of digits. A dual-channel tape-recorder with stereo-phonic ear-phones was used for the test. Digits were presented through these ear-phones in groups of six in such a way that half the digits came to the left ear, the other half to the right. After each group of six numbers, the subject reported everything the he/she heard, in any order he/she liked. They were 32 groups of six digits, making a total possible score of 96 for each ear. The result was 90.25% for the left ear and 92.25% for the right ear.

1.10 Summary

This chapter begins by outlining the motivation for carrying out the research reported herein. The physiology of the human speech production organ is then discussed. This is followed by a discussion of the classification of speech sounds and their phonetic transcription. This leads on to the formulation of the source-filter model of human speech. The other organ which is vital to human communication via speech is the ear, which is involved in hearing. Hence, the parts of the ear are discussed as well. The way in which human perceived speech sounds has also been dealt with. In order to put into context, the capability of the computer speech recogniser (which will be discussed

later) that I have implemented, the human capacity for the perception/recognition of speech has also been discussed.

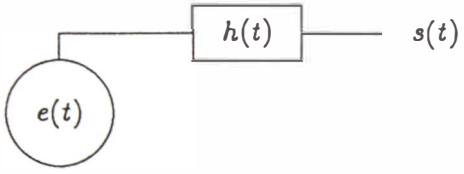


Figure 1.8: *The source-filter model of speech production.*

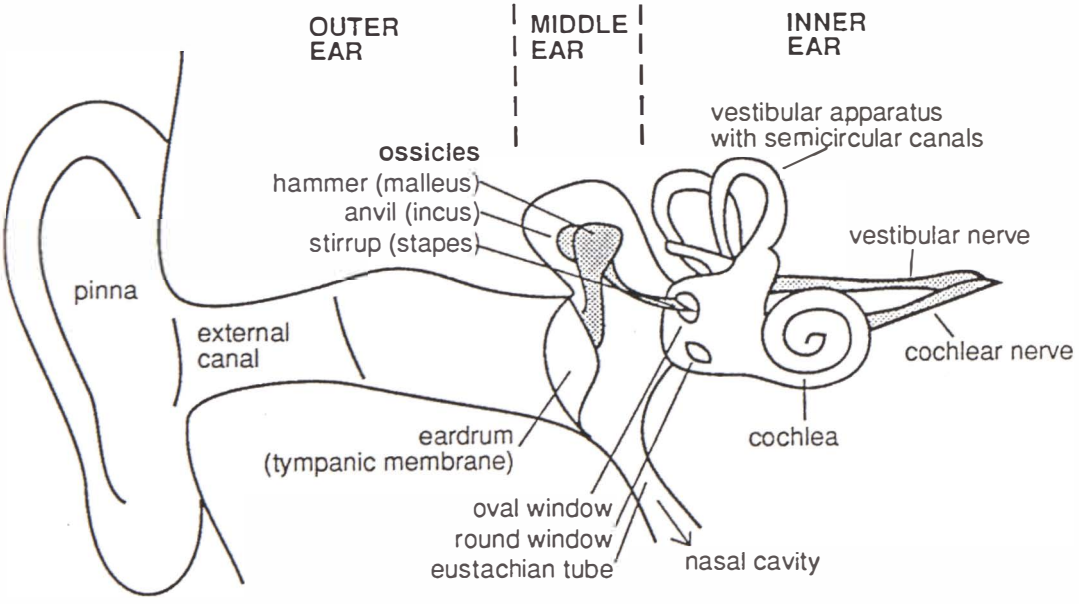


Figure 1.9: *Schematic diagram of the human ear. After Flanagan (1972).*

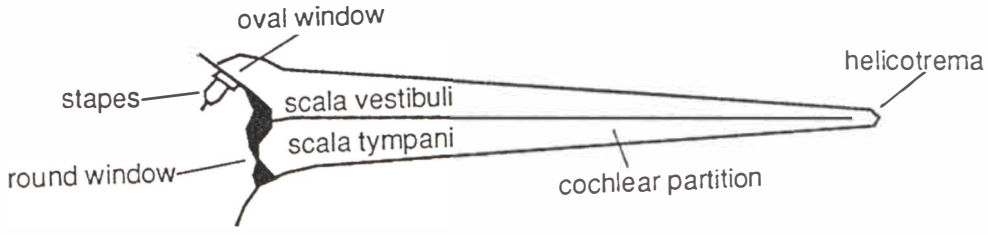


Figure 1.10: *Simplified diagram of an uncoiled cochlea.*

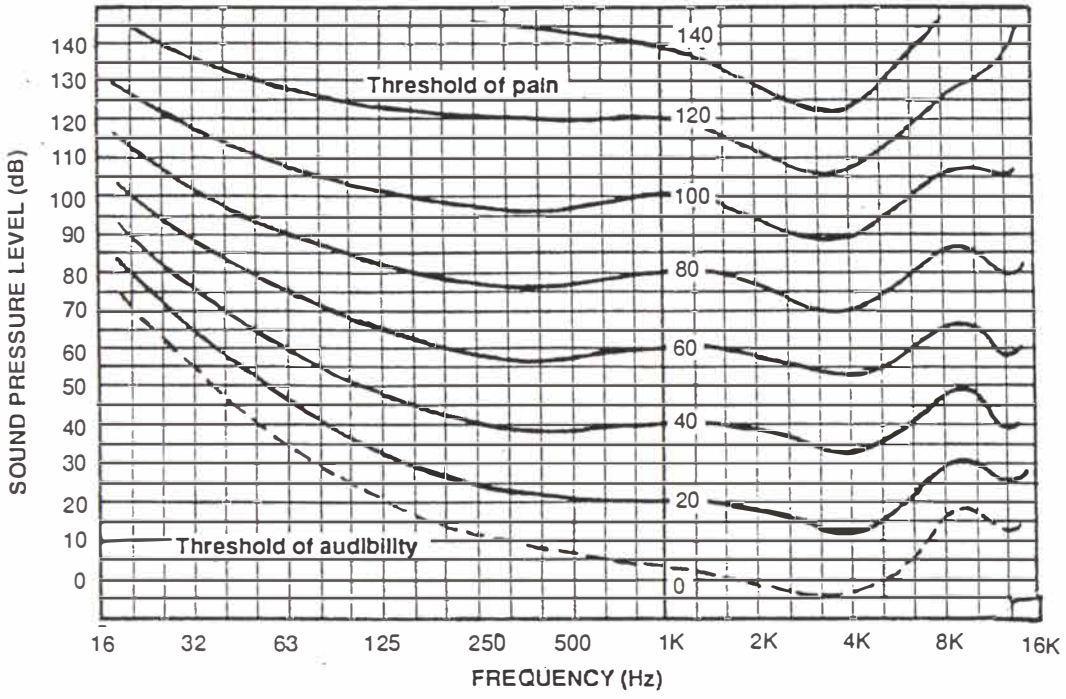


Figure 1.11: *Contours of equal loudness (phons). For example, a 63-Hz tone with a sound pressure level of 75 dB and a 250-Hz tone of 58 dB, a 1000-Hz tone of 60 dB and a 4000-Hz tone of 53dB sound equally loud because they all lie on the 60-phon equal loudness curve. After Hills and Shipley (1985).*

Loudness in decibels	Representative sounds
120 - 140	Jet take-off Artillery fire Riveting
100 - 120	Sonic boom Orchestra music fortissimo Rock band
80 - 110	Truck unmuffled Loud street noise Police whistle
60 - 80	Noisy office Quiet typewriter Average radio
40 - 60	Noisy home Average conversation Quiet radio
20 - 40	Private office Quiet home Quiet conversation
0 - 20	Rustle of leaves Whisper Human breathing

Figure 1.12: *Sound pressure levels of representative sounds and noises.*

EQUIVALENT LOUDNESS IN TERMS OF

Threshold pressure (0.0002 microbar)	Phons	Decibels
1	0	0
10	10	10
10^2	20	20
10^3	30	30
10^4	40	40
10^5	50	50
10^6	60	60
10^7	70	70
10^8	80	80
10^9	90	90
10^{10}	100	100
10^{11}	110	110
10^{12}	120	120
10^{13}	130	130

Figure 1.13: *Conversion between the decibel scale and the phon scale for loudness measure. After Hills and Shipley (1985).*

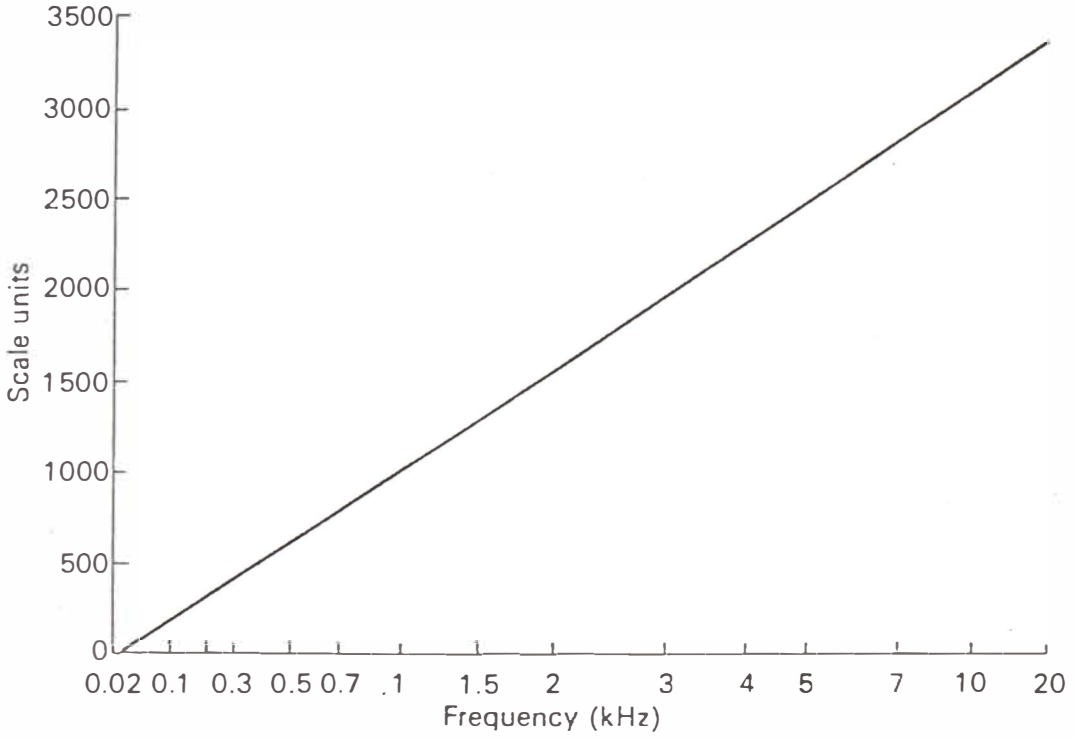


Figure 1.14: *Conversion of mel scale to frequency scale in Hz. After Lieberman and Blumstein (1988).*

Chapter 2

SPEECH PROCESSING - A REVIEW

"Copying from one is plagiarism, copying from many is research."
(Anonymous)

The term "speech processing" is used throughout this thesis to mean the automatic recording, storing, manipulating, transforming and replaying of human speech, using analogue and/or digital electronic devices and all kinds of computer hardware, assisted by sophisticated software. From an engineering point of view, speech processing research falls into four areas: speech synthesis, speech analysis, speech coding and speech recognition. There are a lot of overlap between these four areas, and development in one area often has impact on the other areas. In the following sections, the historical developments of these four areas of speech processing are recounted. From these historical development, the interplay between these four areas will become clear.

2.1 History of speech synthesis

The ability of human beings to speak is unique among all the life forms and this has often been considered as clear evidence of our semi-divine nature. While stories of animals conversing with humans are generally thought of as fairy-tales, it is perhaps for this reason that legends of gods speaking to humans are widely believed or, if not, at least invested with mystical or religious significance. Ancient priests siezed this opportunity to enhance the stature of their gods by making their idols speak directly to the people. Talking statues, miraculous voices, and oracles were well known during the Greek and Roman civilizations. By fitting cleverly concealed tubes to the mouths of the statues, a hidden priest could then speak through the tube (Wheatstone, 1879).

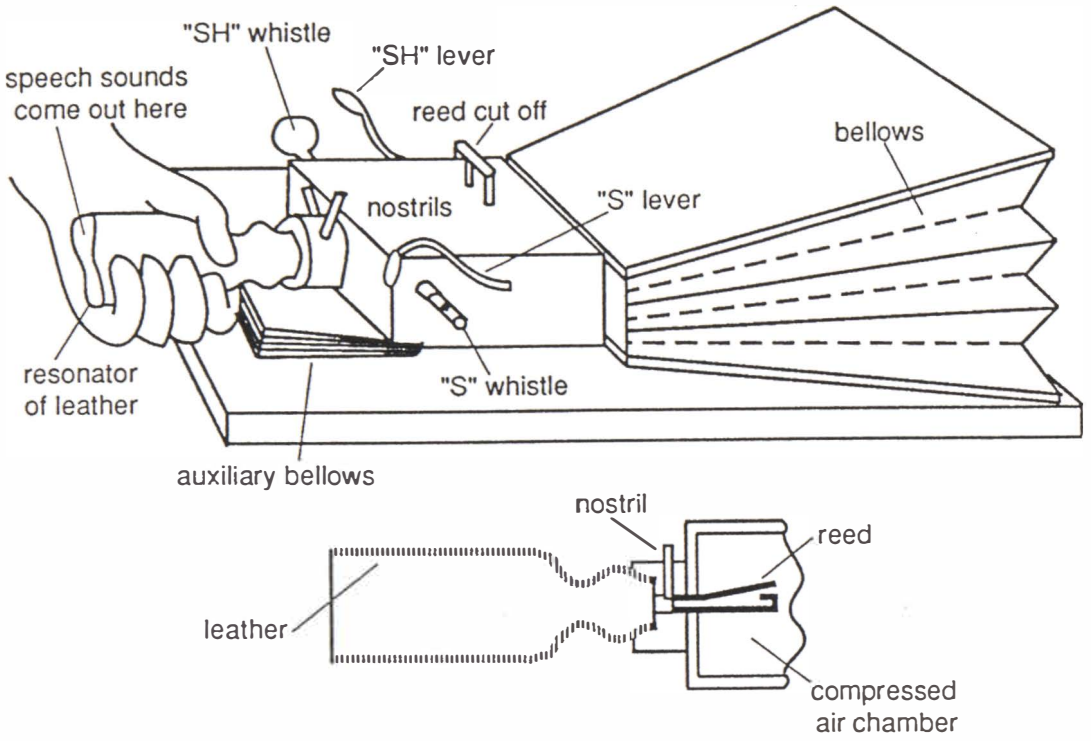


Figure 2.1: *Wheatstone's reconstruction of Von Kempelen's speaking machine.*

Although these speaking tubes can hardly be considered speech synthesizers, they do indicate that even in those ancient times, human beings felt the need, even if they did not possess the technology, to synthesise speech. Despite the advanced state of Greek science and Roman engineering, very little progress in the study of speech was made during the classical times. As superstition yielded to the scientific approach during the Renaissance, speech became a respectable field for scientific research. Early attempts were made to imitate human voices. These efforts invariably took the form of mechanical devices (Flanagan, 1972b). This contrasts with modern speech processing research which, almost without exception, is implemented electronically and, more recently, in very fast digital signal processing chips.

2.1.1 Mechanical speech synthesisers

One of the earliest documented experimental researches into speech synthesis was made by Wolfgang Von Kempelen of Vienna (Paget, 1930). His apparatus (see Figure 2.1) consisted of a kitchen bellows which supplied air to an enclosed box fitted with a bagpipe reed. The reed in turn excited a single hand-held leather resonator that emitted vowel sounds and many consonants. A few years later, in 1791, he built and demonstrated a more sophisticated machine which could speak whole phrases in French and Italian (Dudley and Tarnoczy, 1950). In the same year, he wrote a book on the mechanism of speech and the construction of speaking machines (Kempelen, 1791). He tells of how he arrived at his final design, how the instrument is played, and the discoveries he made concerning the nature of speech.

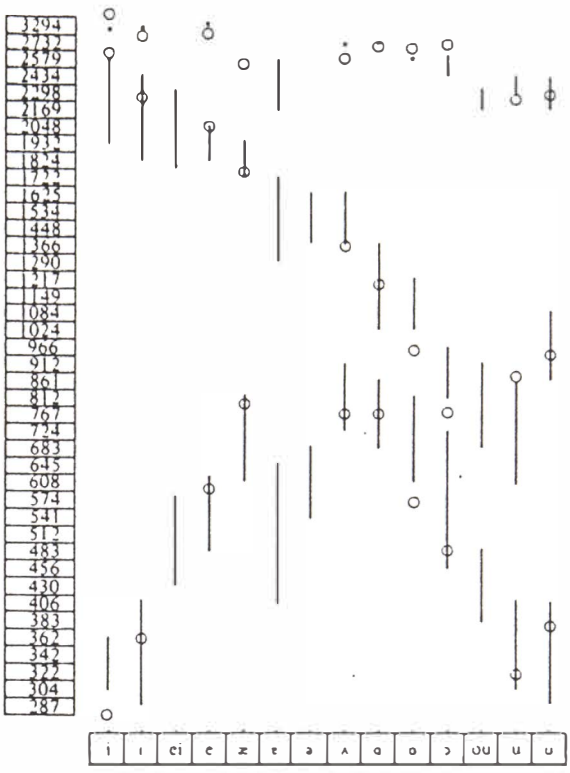


Figure 2.2: *Resonance chart constructed by Sir Richard Paget. The vertical bars are the ranges of resonant frequencies of the various vowels while the circles are typical formant frequencies measured by modern instruments.*

The book was published simultaneously in German and French (Linggard, 1985).

In 1837, Sir Charles Wheatstone constructed an improved version of Von Kempelen's machine based on the latter's description (Bell, 1922). It is interesting to note that during Alexander Graham Bell's boyhood in Edinburgh, Scotland, he was lucky enough to see a demonstration of Wheatstone's construction. Greatly impressed with the device, Bell set out to construct a talking toy of his own. With advice from his father and assistance from his brother, Bell made a cast from a human skull and moulded the vocal parts in *guttapercha* [1]. He managed to turn out vocal parts which represented the lips, tongue, palate, vocal-chord orifice, teeth, pharynx and velum (Bell, 1922). These parts were activated by levers controlled from a keyboard. The device could be manipulated to produce a few simple utterances, apparently well enough to attract notice from the neighbours (Flanagan and Rabiner, 1973).

Some of the most remarkable and extensive experiments in speech synthesis were conducted by Sir Richard Paget (1930). Although the use of electronic instruments was then becoming widely available as tools for speech processing, he chose not to use these instruments. Instead, using his own finely-tuned sense of hearing and intuition, he constructed a resonance chart of the vowels. The chart is depicted in Figure 2.2. From this chart, it can be seen that Sir Richard identified two main resonances

[1] Tough greyish-black plastic substance got from latex of various Malayan trees. Derived from the word *getah* which means gum and *percha* which is the name of a tree (Sykes, 1976).

(formants) for all the vowels, and for some, a third resonance. For comparison, the resonant frequencies of the same vowels measured by modern speech processing techniques are superimposed in the same figure. It is quite amazing that Paget's resonance chart, constructed without any instruments other than his own ears, were quite accurate compared to that which can be constructed from data obtained from modern apparatus and measurement techniques.

Through extensive experimentation, Paget also taught himself to manipulate the resonant frequencies of his own vocal tract, and he could vary the first and second formants almost independently. He noted that the position of the tongue, the size of the gap left by the tongue, and the lip were significant in determining the resonances of the vocal cavity. He tested his theory by making plasticine models of the vocal tract and performed recognition tests before an audience interested in phonetics. He found that all the sounds were correctly identified by a majority of the listeners and thus his theory was verified.

He then proceeded to investigate the acoustic properties of resonant cavities. He realised that cavities may be connected in parallel, or in series, and that the parallel connection was to be preferred because it allowed the cavities to be tuned independently of one another. In his later study, he discovered the extra, nasal resonance and rightly concluded that it was due to air resonating in the nasal cavity. He also studied the vibration of the vocal cords, and noticed that their frequency of vibration varied with changes in air pressure and the tension in the cords. Drawing from his vast accumulated knowledge on speech synthesis, he designed an interesting car horn which shouted "away! away!". By "conducting" an "orchestra" of seven assistants, each armed with one of his patented resonators which emitted individual sounds, Paget distinguished himself by producing a complete, synthetic phrase which said "oh mother are you sure you love me."

2.1.2 Electrical synthesisers

The first electrical synthesizer was made by Stewart (1922). Figure 2.3 is a schematic diagram of his invention which consisted of two resonant electrical circuits excited by current supplied from a battery source. The current could be switched on or off either by a motor-driven interrupter or a manually operated buzzer. He observed that appropriate tuning of the resonant circuits resulted in the production of sounds that approximated all the various vowels and semi-vowels. He noted that the frequency of interruption determined the pitch of the vowel; but not the type of vowel that was produced. This agrees with current understanding of the human voice production mechanism. The frequency of vibration of the vocal cords is known to determine the pitch, while the adjustment of the vocal tract shape is known to characterise the vowels (see §1.3 and §1.4).

Stewart's synthesiser could also generate dipthongs. This was achieved by adjusting the resonant circuits rapidly, in order to shift from the initial to the final vowel sound of the dipthong pair. He also produced some of the fricative consonants by setting the resonant circuits at higher frequencies than that for the vowels and semi-vowels, and by forcing the excitation current to consists irregularly spaced short pulses.

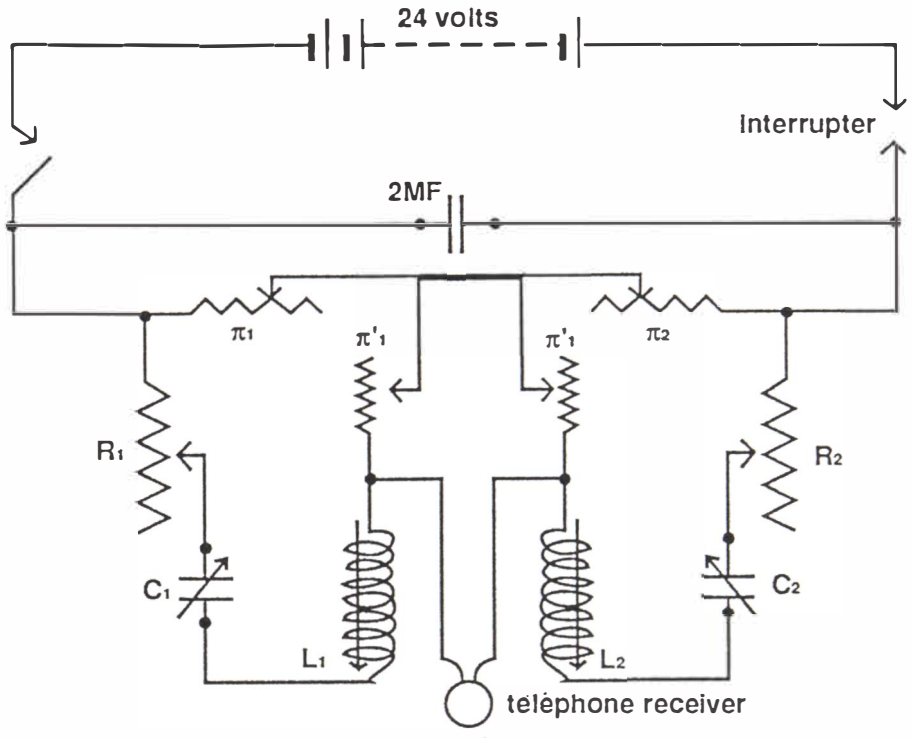


Figure 2.3: *The first electrical synthesizer constructed by Stewart in 1922.*

Dudley, Riesz and Watkins (1939) successfully demonstrated another speech synthesiser at the New York World Fair. A schematic diagram of the device, which was called Voder (for Voice Operation DEmonstratoR), is depicted in Figure 2.4. It consisted of ten contiguous bandpass filters connected in parallel. These filters were excited from either a random noise source or a simulated glottal pulse source. The excitation source was selected by manipulating a wrist bar while the pitch of the glottal pulse was controlled by a foot pedal. The output of each filter was then passed through a potentiometer which could be varied by operating a key with a finger. These weighted filter outputs were subsequently added together to produce the desired sounds. Three additional keys provided transient excitations of selected filters to simulate stop sounds. The Voder was "played" like an organ or piano and it took about a year to train a "performer". Dudley *et. al.* (1939) reported that a trained operator could use the machine to produce "intelligible" speech from prepared scripts.

Following the success of the Voder, other electronic synthesisers soon followed. These are mostly based on the same principles as the Voder. The speech synthesisers of Munson and Montgomery (1950) and Cooper *et. al.* (1950) are typical examples. Strictly speaking, these synthesisers should be termed speech analyser-synthesisers simply because these two components are built into a single unit. The analysis is usually accomplished by splitting the speech into various frequency bands. The energy and the frequency of zero-crossing in each band are then measured by electronic circuits. In addition, the fundamental pitch is also measured. These parameters are subsequently transmitted to a synthesising circuit which reconstructs the speech.

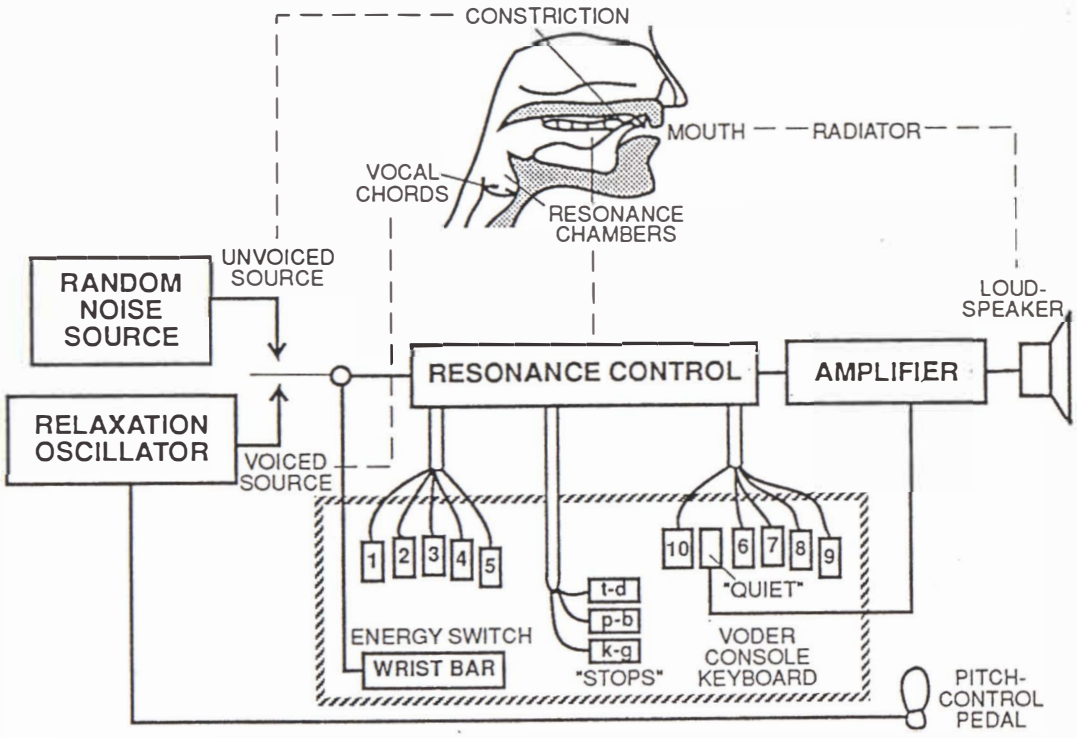


Figure 2.4: *Schematic diagram of Voder. The wrist bar selects a source to excite the filter banks while the foot pedal controls the pitch of the glottal pulse source. Each key operate a potentiometer controlling the output of each filter. These weighted outputs are added together to produce the desired sounds.*

Homer Dudley (1950) actually built and patented a speech analysis and synthesis system called Vocoder (VOice COder-decoDER). The critical distinction between speech synthesisers of the Voder's generation and those of the Vocoder's generation is that the former are controlled by hand-operated keys while the latter are not. An advantage of Vocoders is that the precision of the control parameters (thus the quality of the synthesised speech) is not dependent on the skill of trained "performers".

2.1.3 Digital synthesisers

Towards the end of the 1940s and early 1950s, the emphasis on the practical implications of the sampling theorem by Shannon (1949), and the appearance of digital computers, opened up new avenues for speech processing research in general and speech synthesis in particular. By this time, speech synthesisers capable of generating speech in foreign languages such as Japanese (Oizumi and Kubo, 1954) and German (Winckel, 1951) began to emerge. The sampling theorem tells us how to accurately represent continuous physical signals, such as speech, by discrete samples. Digital computers are able to store large quantities of data, and to perform complicated mathematical operations (*e.g.* Fast Fourier Transform) on these data very accurately and at great speed. These developments mean that new methods of synthesising speech can be simulated, evaluated and optimised without having to build electronic synthesisers.

The 1960s marked the beginning of the polarisation of two different approaches to the design of speech synthesisers. The first approach derives parameters from the the speech signal alone while the second models the speech signal as an integral part of the human speech production mechanism.

The first approach leads to what is often known as the "signal model" (Linggard, 1985). This is the simpler approach because it does not assume any knowledge of the speech production mechanism. Thus, speech synthesisers of the signal model type are easier to design and construct. These synthesisers are similar in design and construction to that of the Vocoder. They all comprise three or four resonant circuits to simulate the effects of formants (Gold and Rader, 1967; Gold and Rabiner, 1968).

The second approach, sometimes called the articulatory or system approach (Linggard, 1985), is much more difficult. However, considerable progress has been made by correlating X-ray photographs of human vocal tracts with speech sounds. Rabiner (1968), and Flanagan and Landgraf (1968) have simulated such speech synthesisers on digital computers.

In their frequently cited paper, Atal and Hanauer (1971) broke new ground by introducing a novel technique for analysing and synthesising speech. Called Linear Predictive Coding (LPC), this has become the standard technique in speech processing research. Because of its importance, a full treatment of this technique is presented in Chapter 3.

During the 1970s, advances in speech synthesis began to find applications in the teaching of phonetics (O'Malley and Kloker, 1971), in giving deaf people "visible" speech and dumb people voices (Nakano *et al.*, 1970), and in the wiring and fabrication of telephone apparatus (Flanagan *et al.*, 1972). The last application is

remarkably successful. Here, a computer automatically converts printed wiring instructions into synthetic speech, thus freeing the technician's eyes and hands, to prevent them being diverted from the apparatus being worked on. Rabiner *et. al.* (1972) tested the system on a production line and found that no errors were made in wiring crossbar-4 telephone equipments. Speech synthesis techniques have also been implemented in digital (Rabiner *et al.*, 1971) and transistorized (Ptacek, 1972) hardware to give voices to computers (Flanagan, 1972b; Trupp, 1970).

By 1973, the field of speech synthesis had made significant progress and there already was a large amount of literature on the subject. To provide "trails" for backtracking the literature, Flanagan and Rabiner (1973) carefully selected and assembled a total of 46 technically important papers that accurately marked the major stages in the development of speech-synthesis technology. Advanced students who are new to the field should be able to digest the papers in a relatively short time. Some of the papers also describe projects which should be stimulating to and are certainly within the capabilities of undergraduate students of electronic engineering.

The speech group at the University of Canterbury was also active in speech synthesis research during the 1970s. Neoh (1973) designed and built a resonance synthesiser using CMOS devices. From an input string of phonemes and a look-up table, a set of control signals for the synthesiser were calculated by an EAI-640 digital computer. These control signals were then fed to the analogue circuits via an interface. Tucker *et. al.*(1977), Lamb and Bates (1978), and Tucker and Bates (1978) invoked such speech synthesis technology to develop an interactive system for assisting music students and their teachers.

2.1.4 The present

Speech synthesis is at present an active field of research. This is evidenced by the amount of literature on the subject. In 1981-1984 alone, some one thousand papers are found to be classified under "speech synthesis" in IEE (1985). In addition, commercial speech synthesisers that can be integrated into personal computers have appeared (McLaughlin, 1981; Bruckert, 1984; and Gunawardan, 1987).

McLaughlin (1981) of General Instrument Corporation, has introduced a single chip speech synthesis system that contains a digital parametric synthesiser, a 4-bit control processor, 16 k-bits of ROM, and a digital-to-analog converter. The device, designated the General Instrument SP0256, is capable of generating clear, understandable speech which retains the natural inflection and intonation of the original speaker. In addition, an on-board controller allows the synthesiser to operate at a wide range of bit rates so that trade off may be made between memory requirements, voice quality, and vocabulary size in individual applications.

Bruckert (1984) of Digital Equipment Corporation (DEC) has introduced a new product called DECtalk. It converts standard ASCII text into "human-sounding, highly intelligible speech". Thus, in contrast to a video terminal which communicate through the sense of sight, DECtalk represents a new kind of terminal which communicate through the sense of hearing.

More recently, R. Gunawardan (1987) has categorised the details of a range of

speech synthesis devices which are being introduced by Texas Instruments Limited. The product range includes four voice synthesis devices and two dedicated voice ROM devices. These devices are designed for two particular attributes – namely increased flexibility of design and system integration, and the capability to produce better quality voice output.

On a less serious side, it is interesting to note that speech synthesis technology has also been employed by the advertising industry (O'Connor, 1990). A talking advertisement has appeared in a magazine in the United States. The heart of this advertisement is a computer chip which stores the parameters from which a sentence can be synthesised. The talking advertisement was manufactured by Texas Instruments Australia. The whole package including the amplifier circuitry, miniature battery and printing costs less than $Aus10.00. The advertisement is activated by lifting a "lift me" label. Once activated, it gives a 42 word sales pitch that can be played 650 times before the battery runs out.

From what has been outlined so far in this chapter, it can be seen that the knowledge of speech synthesis has grown from a laboratory curiosity to something of considerable importance. However, to conclude this section, it is only fitting to quote (with due apologies) Professor Stephen W. Hawking (1987)[2]

> *With this (a communications programme called Living Center) I can both write books and papers, and speak to people using a speech synthesiser ... and a small personal computer mounted on my wheelchair. The speech synthesiser has made all the difference: In fact, I can communicate better now than before I lost my voice.*

2.2 History of speech analysis

2.2.1 Introduction

Speech analysis is essentially the implementation of algorithms which process a speech waveform into useful (to a particular application) parameters. At this level, it is simply the analysis of a time-varying waveform. No knowledge about the production, articulatory dynamics or the phonetic descriptions of the speech being analysed is assumed. However, additional knowledge (especially of the type described in the last sentence) about the speech being analysed affect certain aspects of analysis and the form of the algorithms, for example in the choice of sampling frequency and the rate at which analysis should be carried out.

Speech analysis plays a central role in many speech processing applications. Speech synthesis, speech recognition and speech coding are examples where other speech analysis techniques are used. In all these applications, a speech analyser is used as a front-end processor, in order to extract a set of parameters from a segment of speech being analysed. In speech synthesis, these parameters are used to

[2]Stephen Hawking holds Newton's chair as the Lucasian Professor of Mathematics at Cambridge University. In 1985, he had a tracheostomy operation because of pneumonia. The operation removed his ability to speak. He is widely regarded as one of the most brilliant theoretical physicist since Einstein.

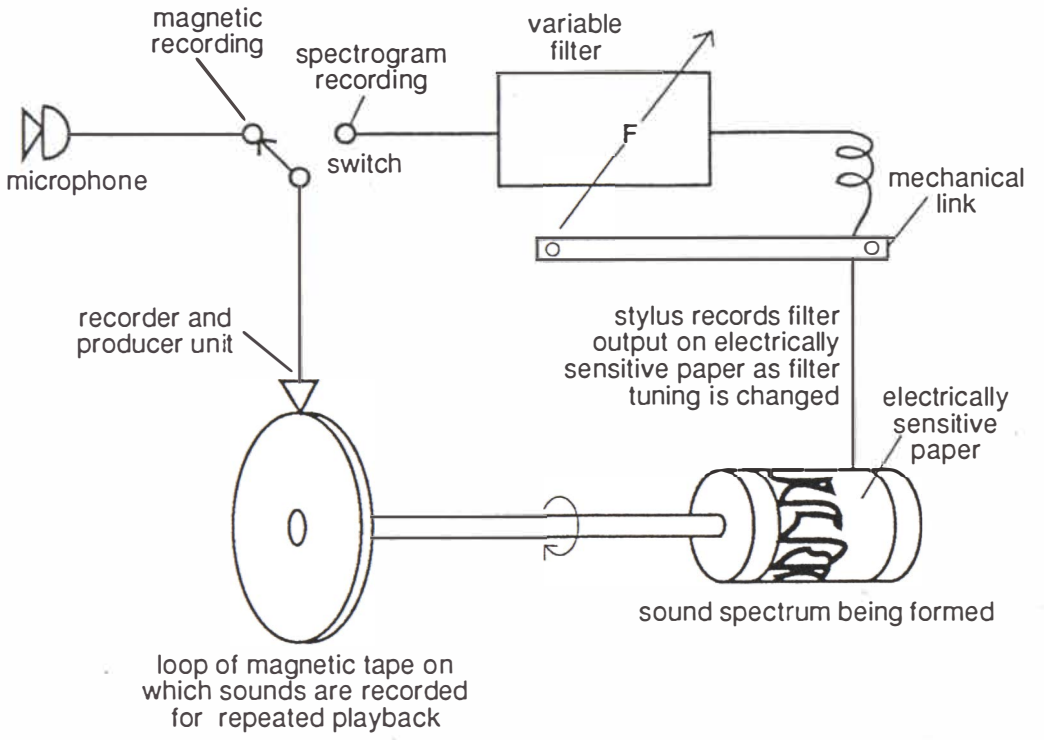


Figure 2.5: *Schematic diagram illustrating the principal components of the sound spectrograph.*

reconstruct the speech; in speech recognition, they are used as inputs to the speech recogniser; and in speech coding, they allow the speech to be digitally encoded with lesser number of bits than that required to directly encode the digitised samples.

Unlike Paget (1930) who had the rare ability to analyse speech signals using only his finely-tuned ears, most people have to rely on some sort of instruments and/or computational techniques to assist them in analysing speech waveforms. In the following sections, the historical development of these instruments and/or computational techniques are summarised.

2.2.2 Frequency domain analysis

The sound spectrograph (Koenig *et al.*, 1946) was the first speech analyser that perform frequency domain analysis. A schematic diagram illustrating the principal components of the sound spectrograph is shown in Figure 2.5. The speech to be analysed is first recorded on magnetic tape. It can hold up to 2.4 seconds of speech. Connected to the output of the magnetic tape recorder unit is a band pass filter. Its bandwidth can be set at either 45Hz (narrow band) or 300Hz (wide band), while the centre frequency of the pass band can be varied from d.c. to 8kHz with an increment of 15Hz. To analyse the speech, the recording is repeatedly played back as the variable frequency band pass filter scans through the entire speech frequency bands. The output of the filter energises an advancing stylus which writes on a piece of electrically sensitive paper mounted on a rotating drum. Thus, line by line, a

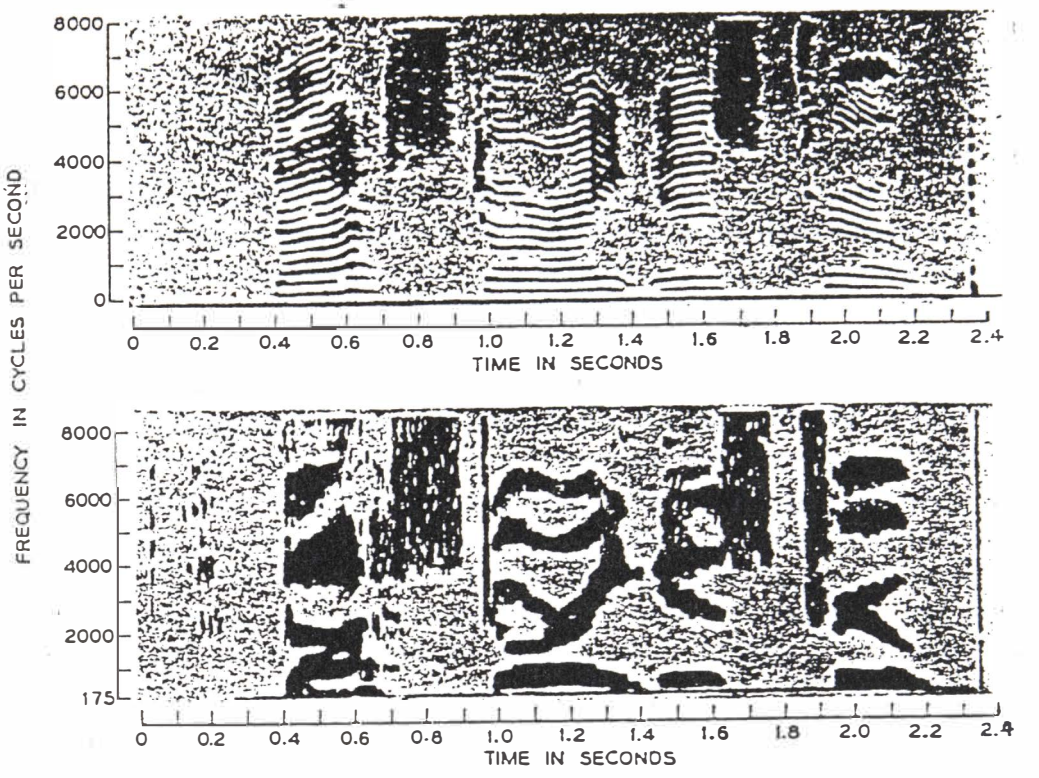


Figure 2.6: *Narrowband and wideband spectrograms for the spoken phrase,* The story is true. *After Dudley (1955)* .

spectrogram is produced.

Figure 2.6 shows two examples of spectrograms; one using a narrow band filter and the other a wide band filter. They show the energy content at a particular frequency and a particular instant in time. Specifically, the darker the spectrogram is at a particular point means the higher the energy content at that point. From Figure 2.6a, one sees that a narrow band filter produces horizontal striations in the spectrogram, indicating better frequency resolution. In the case of a wideband filter, the frequency resolution is poorer, thus causing the horizontal striations to merge together to form four or five distinct dark bands (see Figure 2.6b). These dark bands are called formants, and they indicate where (with respect to frequency) most of the energies are concentrated. The spacing between the vertical striations indicates the vocal cord vibration period. From Figure 2.6b, one also notices that for unvoiced speech, most of the energies are concentrated at frequencies above 4 kHz, whereas for voiced speech, they are concentrated at frequencies below 4 kHz. Therefore, as we have seen, spectrograms reveal a lot of information about the speech being analysed. For this reason, they remain widely used.

By 1970, many speech processing systems were based on small or medium-size computers (Oppenheim, 1970), thus providing mechanisms for carrying out sophisticated studies in speech analysis (Denes and Mathews 1970; Schroeder 1969). With such a facility, it has become convenient to generate spectra digitally rather than by making an analog magnetic tape recording, which is then analysed off-line by a

spectrograph machine. Furthermore, it is often advantageous to closely (in time) relate time waveforms and spectral waveforms and to be able to choose bandwidths flexibly during the analysis.

Although the discrete Fourier transform (DFT) algorithm (Witten, 1982) can be used to compute the spectra of speech waverforms, it involves too many computations to be practically useful. In fact, to calculate the spectrum of N samples of speech using the DFT takes N^2 operations, where each operation incurs a multiplication and an addition of complex numbers. The fast Fourier transform algorithm (Cooley and Tukey, 1965; Bergland, 1969; Brigham, 1974; Kay and Marple, 1981) reduces this to $N \log_2 N$ operations. This saving in the number of operations means that the advantages of generating spectrograms on a computer can be realised.

Oppenheim (1970) remarked that "At present, it does not seem to be efficient or advantageous to carry out *routine* spectrographic analysis of *large* amounts of speech data on a sophisticated computer facility. However, as the cost of small computers and digital hardware decreases, it may eventually be practical and economical to reserve a small facility for preliminary analysis of speech signals, including displays of spectra and time waveforms." Twenty years later, we see that his remark has been vindicated. There are now a wide range of DSP chips available for signal processing (Aliphas and Feldman, 1987). A very popular DSP chip is the TMS320 series produced by Texas Instruments. The Canterbury University Computer Aided Speech Therapy Tool (CASTT) is based on the Texas Instrument TMS320C10 digital signal processing chip. The CASTT is capable of generating a four-colour spectrogram on a computer screen in real time (Watson *et al*, 1988; Bates *et al*, 1987).

2.2.3 Cepstral domain analysis

Cepstral domain analysis is another technique for analysing speech signal. A speech signal, $s(t)$, can be modelled as a convolution of an excitation signal, $e(t)$, and a vocal tract filter, $v(t)$ (Fant, 1973). Mathematically, this can be expressed as

$$s(t) = e(t) \odot v(t) \tag{2.1}$$

where $\odot$ is the convolution operator. Taking the Fourier transform on both sides yields

$$S(f) = E(f) \times V(f) \tag{2.2}$$

where $\times$ is the multiplication operator, f is the frequency. In Equation (2.1) and Equation (2.2), it should be noted that the lower case letter and the upper case letter represent the time domain and Fourier domain respectively. The logarithm of the transform is then calculated:

$$\log(S(f)) = \log(E(f)) + \log(V(f)) \tag{2.3}$$

Finally, the inverse Fourier transform is computed:

$$\mathcal{F}^{-1} \log(S(f)) = \mathcal{F}^{-1} \log(E(f)) + \mathcal{F}^{-1} \log(V(f)) \tag{2.4}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform operator. The result, the spectrum of the log of the frequency spectrum, is called the "cepstrum" (an anagram of spectrum). The horizontal axis of a cepstrum, which has the dimension of time, is called "quefrency" (an anagram of frequency) (Noll, 1964).

From Equation (2.4), it can be seen that the result of this complex operation is to separate out the excitation signal from the vocal tract filter function (Gulamhusein, 1973). In other words, the excitation signal is deconvolved from the vocal tract filter function. This process is also explained pictorially in Figure 2.7. Figure 2.7a shows a 512-sample window of a segment of speech. The speech is sampled at 10kHz and multiplied by a Hamming window (Stremler, 1982) of 51.2ms long. The spectrum of the windowed speech is then computed by performing a DFT operation. The log spectrum is subsequently obtained by taking the logarithm of the spectrum and the result is shown in Figure 2.7b. Notice that, in Figure 2.7b, the envelope is slowly varying and superimpose on it is a faster varying fine structures. The slowly varying envelope corresponds to the frequency response of the vocal tract filter while the faster varying structure relates to the excitation signal. The cepstrum is then obtained by taking the inverse Fourier transform of the log of the spectrum and this is plotted in Figure 2.7c. Inspection of Figure 2.7c shows the peak of the excitation signal at $T_0 = 7.9ms$. The part of the cepstrum below T_0 is the impulse response of the vocal tract filter which is separated out by multiplying the cepstrum by a raised-cosine window (Haykin, 1983) which selects the cepstrum below T_0. After the separation, the impulse response of the vocal tract filter is then Fourier transformed to give the smoothed spectrum as shown in Figure 2.7d. Notice that the smoothed spectrum corresponds to the envelope of Figure 2.7a and the four formants F1-F4 are clearly visible.

The above method of computing cepstral coefficients was used by Luck (1969) for a speaker verification system. In this experiment, a genuine speaker was asked to say the word *my* and 16 cepstral coefficients were computed and stored as templates. To test the effectiveness of the system, twenty-three male imposters were asked to say the same word *my* and again, 16 cepstral coefficients were computed. These coefficients were then compared, using a nearest-neighbour classification technique, to the stored coefficients of the genuine speaker. The system was found to have a false acceptance rate of approximately 6%.

The method described above invokes the FFT in order to compute the cepstrum of a speech signal. Atal (1974) described another method of calculating the cepstrum without the need to use FFT. This method computes the cepstral coefficients from the linear predictive coefficients (see Chapter 3) using the following recursive relationships:

$$
\begin{aligned}
c_1 &= a_1, \\
c_n &= \begin{cases} \sum_{k=1}^{n-1}(1-k/n)a_k c_{n-k} + a_n, & 1 < n < p, \\ \sum_{k=1}^{n-1}(1-k/n)a_k c_{n-k} & n > p. \end{cases}
\end{aligned} \tag{2.5}
$$

where c_n, a_n and p are the nth cepstral, linear predictive coefficients and the order of the predictor respectively.

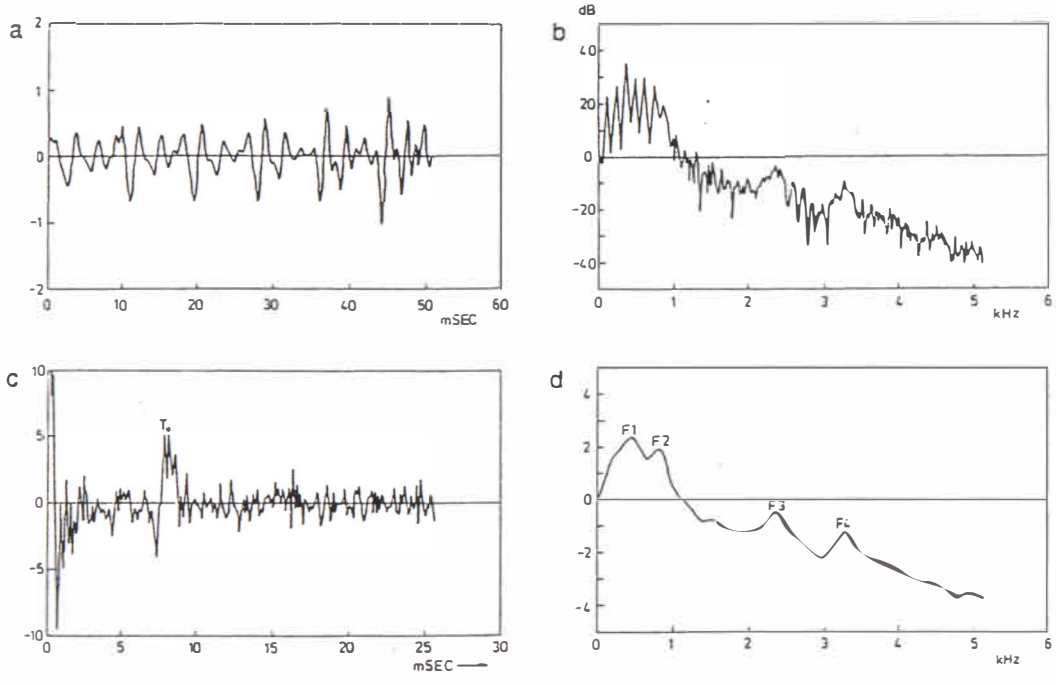


Figure 2.7: *The cepstral technique for speech analysis. (a) The speech waveform of a vowel, (b) The log spectrum of (a), (c) The cepstrum, or inverse Fourier transform of (b), (d) The smoothed spectrum as a result of removing the effect of the excitation signal.*

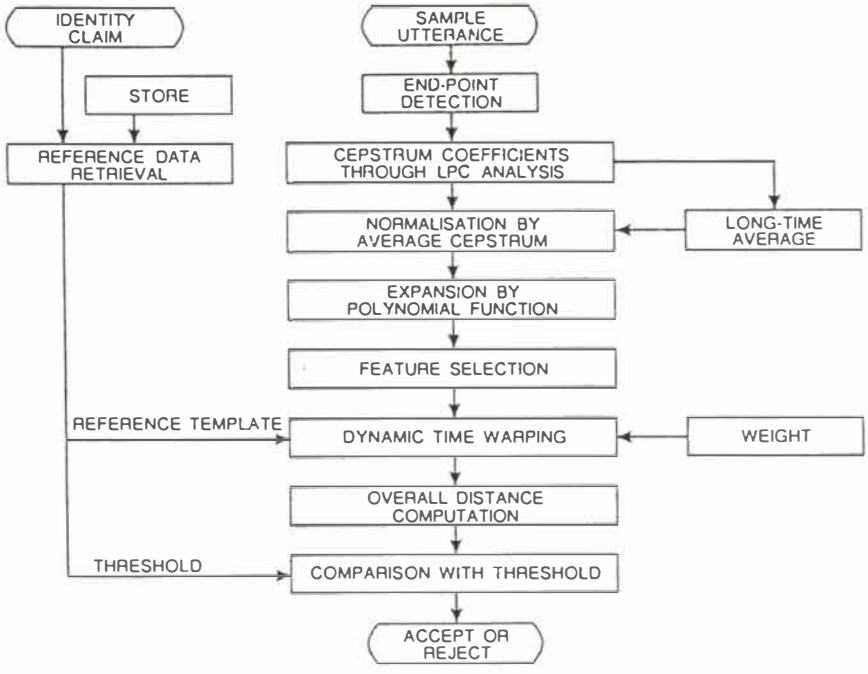


Figure 2.8: *Block diagram of a speaker verification system based on cepstral coefficients as input features. After Furui (1981).*

Furui (1981) studied in detail the capabilities of cepstral coefficients for automatic speaker verification. A block diagram indicating the main parts of the system is shown in Figure 2.8. In Figure 2.8, the identity claim and the sample utterance constitute the two inputs to the system. When a user makes an identity claim by keying in an identification number, previously stored reference data corresponding to the claim is retrieved. The user is then asked to utter a specified sentence. After the endpoints of the utterance are detected, a set of LPCs are extracted from the utterance. The LPCs are computed from 30 ms segments of the digitised speech, with each segment spaced by 10 ms apart.

These LPCs are subsequently converted to cepstral coefficients using the relationship indicated by Equation (2.5). Each cepstral coefficients is averaged over the entire utterance and the average values are subtracted from the coefficients of every frame. From these normalised cepstral coefficients, the mean, slope and curvature (over 9 frames) of each cepstrum coefficients are then calculated. Thus, for each time frame, 30 parameters result. A subset of these parameters is selected for speaker verification, based on the ratio of inter-speaker to intra-speaker variability of each parameters. It is obvious that a parameter with the higher ratio is more effective for speaker verification than one with the lower ratio.

Several sets of utterances were used for the evaluation of the system. These utterances were recorded over a telephone. Both male and female speakers were included in the tests. These tests show that the verification error rate of less than one

percent can be obtained even if the utterances are subjected to different transmission conditions.

More recently, both instantaneous and transitional cepstral coefficients have also been used successfully in speech recognition systems. In particular, Furui (1988) achieved 98% recognition rate with a database consisting of the names of 100 Japanese cities uttered by 20 male speakers.

2.2.4 Formant analysis

Formants are the resonant frequencies of the vocal tract. One of the earliest recorded experiments in formant analysis of speech was conducted by Sir Richard Paget (1930). From the resonance chart that he constructed (see Figure 2.2), he synthesised the English vowels as well as the consonants. It is noteworthy that Paget constructed the resonant chart solely by his ears unaided by any instruments.

While Sir Richard had to rely on his finely-tuned ears to determine the formant frequencies of human speech, modern technology has made the task of formant analysis much easier. The most frequently used technique in the 1970s was the spectrographic technique where a spectrogram is produced. An example of a spectrogram has already been shown in Figure 2.6. It shows the variations of the frequency contents of a speech signal with respect to time. The resonant frequencies can most easily be identified in Figure 2.6. In Figure 2.6, these resonant frequencies show up as horizontal dark bands. Notice that these dark bands only occur in the voiced part of the speech and that there are usually four or five resonant frequencies. These formants are numbered in terms of increasing frequency, for example, the lowest frequency resonance is known as 'formant one' F1, the next as 'formant two' F2, *et cetera.*

There are three main sources of problems with using spectrograms for formant analysis. These are listed below:

- When two formants become close (in frequency) to each other, they merge to form a single band in the spectrogram. This gives rise to problems in numbering and separating the formants. For example, if a spectrogram shows four distinct formants (F1-F4) at the beginning of a phrase, and then F1 becomes so close to F2 that they cannot be distinguished (or separated) in the middle of the phrase, there is no optimal way of deciding this merged band should be labelled F1 or F2. Perhaps a possible solution would be to find out the width of the merged bands and to divide out the band into two at the middle.

- In the case of many vowels, particularly open vowels, a dark band often occurs in the low-frequency region which is caused by the glottal source rather than the vocal tract filter. In other vowels, this peak combines with the first formant.

- In some consonant, particularly nasals, antiresonances (zeros) occur at the same time as the formant resonances (poles). Further, these antiresonances may coincide in frequencies with the formants and so reduce the intensities of these formants so much that the formants are masked.

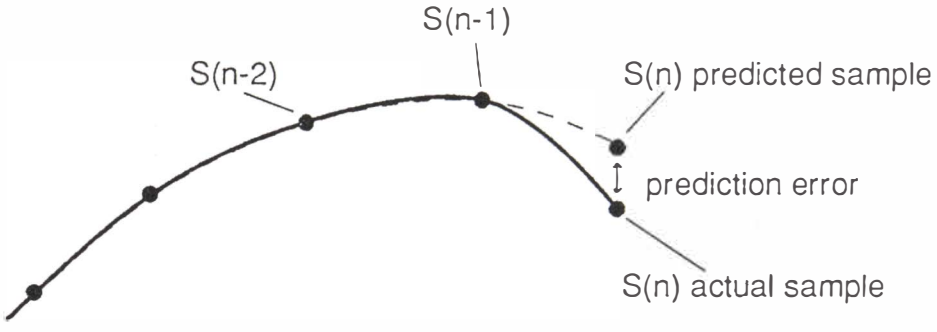


Figure 2.9: *Principles of linear predictive analysis of speech.*

Despite the problems described above, there are algorithms (Markel and Gray, 1976; Ainsworth, 1970; McCandless, 1974; Wood and Pearce,1989; and Fallside and Woods, 1985) which estimate and track the trajectories of formants to varying degree of success.

Wood and Pearce (1989) studied the performance of the excitation synchronous formant analysis technique, particularly where the analysis interval is over the closed phase of the larynx. The improved performance of closed-phase formant analysis is demonstrated by comparison with pitch synchronous and fixed-frame formant analysis. The closed-phase region is determined first using a laryngograph signal and secondly using a modified form of the Gold-Rabiner algorithm (Rabiner and Schafer 1978) fundamental-frequency estimator, using only the acoustic waveform. From the tests that they have conducted, they concluded that the excitation synchronous formant analysis technique is better at tracking the formants' trajectories, with fewer missed or extra formants. Particularly impressive is the technique's ability to follow formant transitions during glides (e.g w,r,l and in voiced segments following plosives).

2.2.5 Linear predictive analysis

The linear predictive technique was first applied to speech analysis by Atal and Schroeder (1967). It was originally used for encoding speech signals efficiently by representing the signals in terms of time-varying parameters related to the transfer function of the vocal tract and the characteristics of the glottal pulse excitation (Atal and Hanauer, 1971).

Figure 2.9 illustrates the principles of the linear predictive analysis of speech where the segment of speech to be analysed is of M samples long. The analysis is carried out by predicting the present sample $s(n)$ by a weighted linear combination of the previous p samples:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{2.6}$$

where $\hat{s}(n)$ is the predicted present sample and $s(n-k)$ is the kth sample to the left of $s(n)$. For example, $s(n-1)$ is the first sample to the left, and $s(n-2)$ is the second sample to the left of $s(n)$, and so on. The weights, a_k, associated with each sample is called the linear predictive coefficients and the order of the predictor is given by p. For example, if $p = 10$ in Equation (2.6), the present sample is predicted by the previous 10 samples, then one says that the prediction is of the order 10 and that there will be 10 corresponding LPCs denoted by a_1 to a_{10}.

As stated in the last paragraph, the linear predictive analysis predicts the present sample $s(n)$ by a weighted linear combination of the previous p samples. Under normal conditions, one expects the predicted sample $\hat{s}(n)$ to be different from the actual sample, $s(n)$. This difference is called the prediction error, and is defined as:

$$e(n) = s(n) - \hat{s}(n) \tag{2.7}$$

where $e(n)$ is the prediction error, $s(n)$ is the actual sample value and $\hat{s}(n)$ is the predicted sample value.

The process is then repeated for all the M samples and the square of the prediction errors are added together. Thus

$$E = \sum_{n=M-p}^{M} e^2(n) \tag{2.8}$$

where E is the sum of the square of the prediction error and p is the order of prediction as defined before.

The goal of the linear predictive analysis is thus to represent the M samples by a set of p LPCs so that the total prediction error E, as defined in Equation (2.8) is minimised over the whole of M samples. Notice also that M should be bigger than p. If the speech signal to be analysed is sampled at 10kHz, a suitable value that is used for M is 200. For a male speaker, this is equivalent to two pitch periods (the average pitch period of a male speaker is about 10 ms). The most frequently used order of prediction, p, varies from 10 to 20.

One algorithm which calculates the LPCs is the Durbin-Levinson algorithm which has already been detailed in book form (Markel and Gray, 1976), tutorial (Makhoul, 1975) and review papers (Schroeder, 1985). The mathematics of the linear predictive analysis will be expounded more rigorously in Chapter 3.

Linear prediction has become the most widely used method of speech signal analysis since its introduction to speech processing. It has been used for speech coding (Atal and Hanauer, 1971), speech recognition (McInnes *et al.*, 1989) and speech synthesis (Trancoso and Tribolet, 1989). LPC analysis has also been used in its original form (Atal, 1974) and in modified form (Lee, 1988). Lee's (1988) method minimizes

the sum of appropriately weighted errors, rather than the sum of the squared errors, as in the original form. The weight is a function of the prediction errors, $e(n)$, with more weight being attached to the smaller prediction errors. Lee (1988) found that the method is insensitive to the placement of the LPC analysis window and to the value of the pitch period. Tests on synthetic vowel data also demonstrate that the robust LPC algorithm is able to reduce the formant and bandwidth error rate by more than an order of magnitude compared to the original LPC algorithm (Lee, 1988).

2.2.6 Shift-and-add (SAA) analysis

Shift-and-add (SAA) is a method of blind deconvolution (Bates and McDonnell, 1989) that was originally developed for deblurring astronomical images (Bates, 1982). It can be used to recover a signal when it has been distorted by an ensemble of random blurring functions. Each blurred signal is shifted until its largest peak is at the centre, and then the ensemble is averaged. The assumption is that the largest peak in each blurred signal is most likely to correspond to the largest peak in the original signal. Thus, by averaging a large number (typically >200) of blurred signals, the original signal is reinforced, while any random or quasi-random contamination tends to cancel out (Minard *et al.*, 1985).

Speech may be viewed as a train of glottal pulses that are filtered by the vocal tract. This can be thought of as a blurring process, where each glottal pulse is blurred by the filter characterising the current state of the vocal tract. The shape of the vocal tract changes as different sounds are uttered, but can be considered to be effectively stationary within any single pitch period (Bates *et al.*, 1987). If the speech is divided into segments of about a pitch period long, and the resulting ensemble of quasi-randomly blurred signals is subjected to shift-and-add, then an estimate of the glottal pulse is generated (Davey and Thorpe, 1987). Since the glottal pulse changes somewhat from pitch period to pitch period, especially as different sounds are spoken and the pitch is altered (Miller, 1959), shift-and-add cannot extract the actual individual pulse shape. Rather it produces an 'average glottal pulse', which is the part of the speech having a constant shape in each pitch period (Davey and Thorpe, 1987). Figure 2.10(a) shows a typical section of a voiced part of a speech waveform spoken by a male. Notice that this section is 52 ms long and there are 5 pitch periods in the section. This means that the pitch period is about 10 ms long. The SAA signal calculated from about 300 pitch periods of voiced speech is shown in Figure 2.10b. Thus, as seen from the last paragraph, shift-and-add provides a non-invasive and non-tactile means for estimating the glottal excitation of voiced speech (Brieseman *et al.*, 1987). A noteworthy feature of this technique is that the estimation (of the glottal excitation) can be achieved in real-time (Watson *et al.*, 1988).

2.2.7 CLEAN analysis

Compression of digitised speech is useful for applications where it is expensive to store or transmit data (Kondoz and Evans, 1988). Examples where this might occur include storing speech samples on a computer system or "talking toy", and trans-

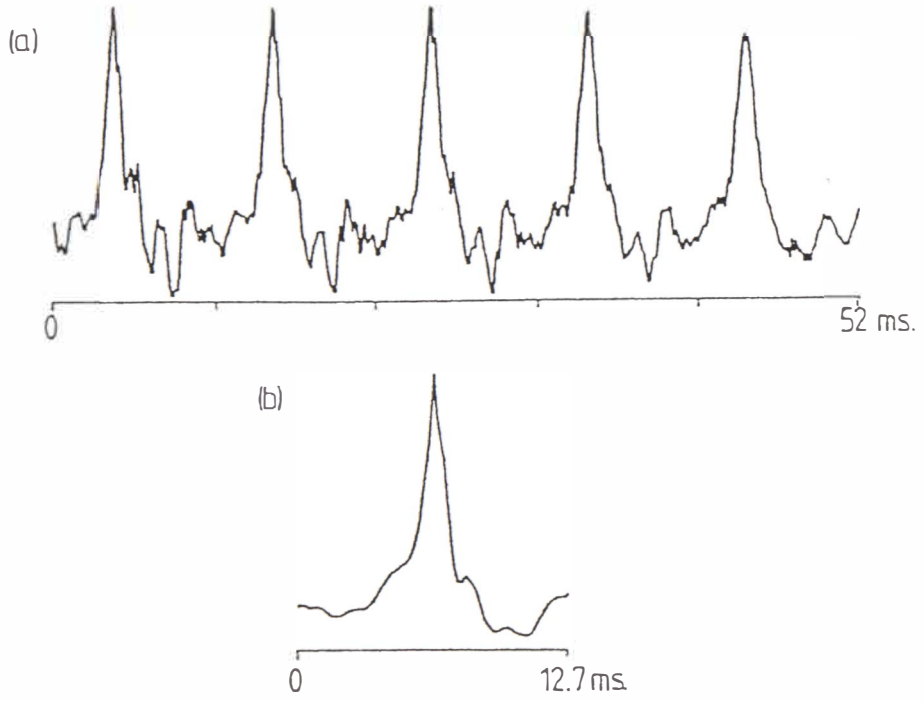


Figure 2.10: *Shift-and-add processing of speech. (a) Typical section (of five pitch period long) of voiced speech. (b) The SAA signal from about 300 pitch periods, which provides an estimate of the average glottal pulse of this speaker.*

mitting speech signals over a bandlimited channel such as a mobile radio link or computer data network. Methods of compressing digitised speech include waveform coding technique such as adaptive pulse code modulation (ADPCM) or sub-band coding, in which the data rate can be reduced to 16-32 kbits/sec (Lafuente, 1983).

The SAA signal can then be used as the kernel in the subtractive deconvolution technique "CLEAN" (Högbom, 1974). As indicated earlier, CLEAN was originally used for deblurring astronomical images, and has been applied in medical applications (Bates, 1981a).

It has been found that applying CLEAN to speech signals creates a "clean" signal (see Figure 2.11) which consists of irregularly spaced pulses. From Figure 2.11c, it can be seen that the clean signal is sparse, that is, it comprises very few non-zero samples. Although the clean signal is sparse, the speech reconstructed by reconvolving the SAA signal with the clean signal has reasonably 'good' quality (Bates *et al.*, 1988).

A large amount of storage (in the case of storing speech) or transmission bit rate (in the case of speech transmission) can be saved by exploiting the sparsity of the clean signal. This is achieved by storing or transmitting the non-zero pulses in the clean signal and the time at which these non-zero pulses occur. Notice that the SAA signal need to be stored or transmitted only once.

Furhter details of the mathematics of the CLEAN algorithm are explained in Chapter 6.

2.2.8 Pulse Code Modulation (PCM) analysis

According to Haykin (1983), pulse code modulation (PCM) system was invented by A. H. Reeves in 1937. The basic elements of a basic PCM system is illustrated in Figure 2.12. The essential operations in the transmitter of a PCM system are *sampling, quantizing* and *encoding.*

In order to ensure perfect reconstruction of the transmitted speech signal at the receiver, the incoming speech signal must be sampled at twice the highest frequency component W of the incoming speech signal in accordance with the sampling theorem (Shannon, 1949). In practice, the incoming speech signal is band-limited by a low-pass filter at the front end of the sampler to make sure that the sampling theorem is satisfied.

After the sampling stage, the speech signal, which is continuous in time, is now represented by a limited number of sampled values per second. However, speech signal has a continuous amplitude range. In other words, within the finite amplitude range of the signal, each sample may assume any of the infinite number of amplitude levels within the finite range. Because the human sense of hearing can only detect finite intensity differences (Haykin, 1983), there is no need to transmit the exact amplitudes of the samples. The implication of this is that the samples may be approximated by a signal consisting of discrete amplitudes selected on a minimum error basis from an available set of finite size. The existence of a finite set of allowable amplitude levels is a basic condition of PCM.

The conversion of a speech sample with a continuous range of amplitudes into one

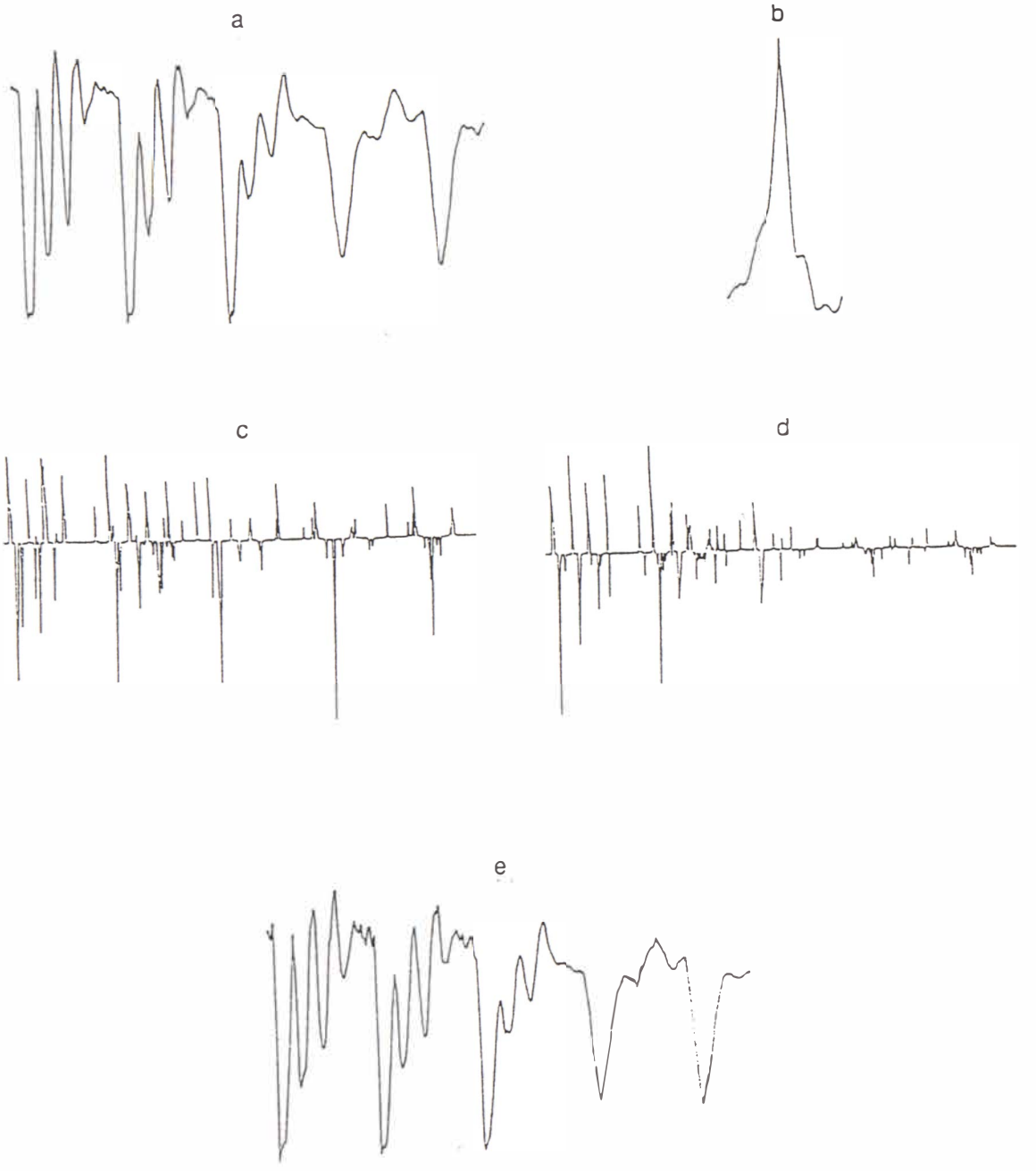


Figure 2.11: *Applying CLEAN to a segment of speech. (a) original speech segment, (b) SAA signal, (c) The clean signal produced from the speech segment and the SAA signal, (d) The clean signal after optimising the amplitudes of the pulses, (e) Synthetic speech reconstructed from the optimised clean signal and SAA signal.*

Figure 2.12: *The basic elements of a PCM system.*

with a finite range of amplitudes is called the *quantization* process. The principle of the quantization process is illustrated in Figure 2.13. Figure 2.13 shows a staircase relationship between the input signal and the quantized signal. Notice that the separation between the quantizing levels is uniform. In speech applications, however, it is preferable to use a non-uniform separation between the quantizing levels. This is because the range of voltages covered by speech signals, from the maximum voltage (corresponding to the loud part of a speech waveform) and the minimum voltage (corresponding to the soft part) is of the order of 1000 to 1. Because the amplitudes are not distributed uniformly, it makes sense to exploit this non-uniform distribution so that the amplitudes which occur more frequently are quantized more finely and vice versa. The end result is that fewer number of quantization levels are required.

The effect of using a non-uniform quantizer is the same as compressing the input speech signal and then applying the compressed signal to a uniform quantizer. Two commonly used non-uniform quantizers are called the μ-law (Smith, 1957) and the A-law (Kaneko, 1970) compressor. The μ-law and the A-law are defined by Equation (2.9) and Equation (2.10) respectively.

$$|v_2| = \frac{\log(1+\mu|v_1|)}{\log(1+\mu)} \tag{2.9}$$

$$|v_2| = \begin{cases} \frac{A|v_1|}{1+\log A}, & 0 \le |v_1| \le \frac{1}{A} \\ \frac{1+\log(A|v_1|)}{1+\log A}, & \frac{1}{A} \le |v_1| \le 1 \end{cases} \tag{2.10}$$

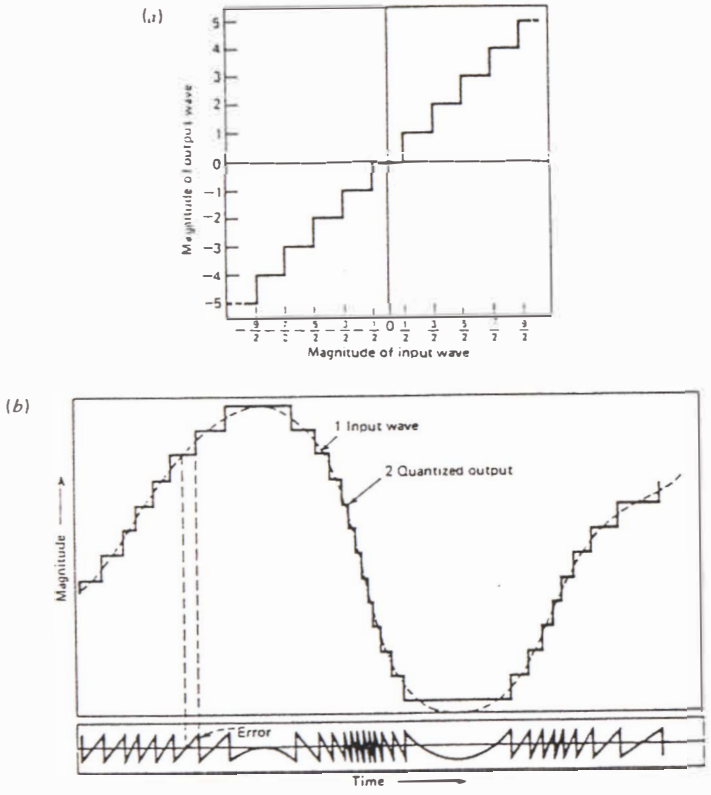


Figure 2.13: *Principle of the quantization process. (a) Characteristic of a staircase quantizer, and (b) An input signal, the corresponding quantized signal and quantization error curve.*

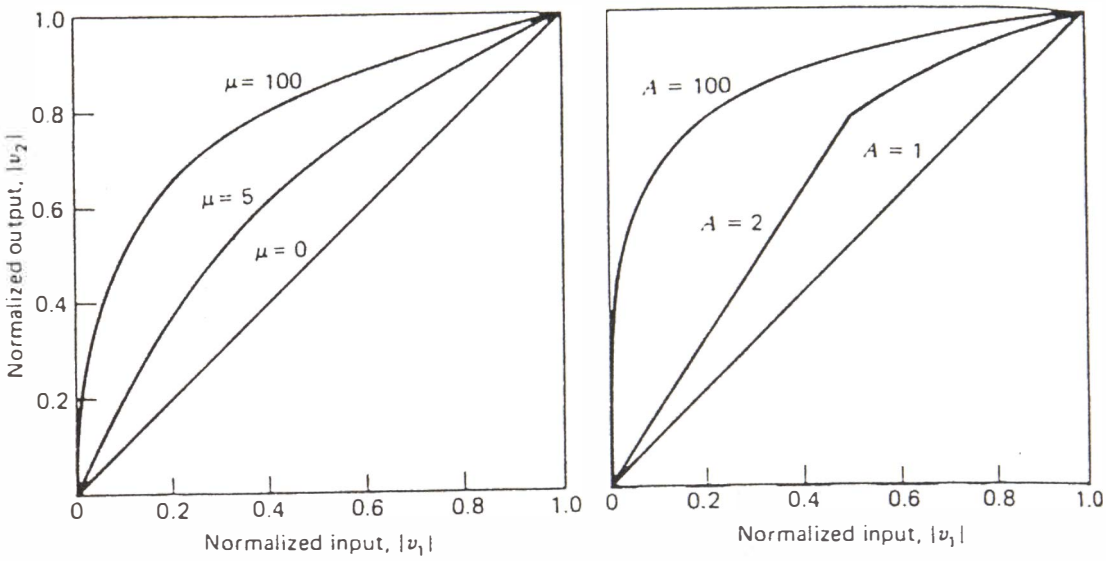


Figure 2.14: *Input-output relationship of μ-law and A-law.*

where v_1 and v_2 are the normalised input and output while μ and A are arbitrary positive constants. In Figure 2.14, the input-output relationships of μ-law and A-law are plotted for three different values of μ and A. Notice that the case of uniform quantizing corresponds to $\mu = 0$ (for μ-law compressor) and $A = 1$ (for A-law compressor). These two compressors are also known as log compressor because they essentially converts the sampled signal from a linear scale into a logarithm one.

After the quantization process, accompanied by the appropriate compression, the speech signal becomes identified with a discrete set of values at discrete instances of time. However, this is still not suitable for transmission over a line or a radio path. This is achieved by the final process involved in the transmitter of a PCM system called the encoding process. In a binary system where there are N (with $N = 2^n$) quantization levels, this means converting each quantized sample into n *bi*nary digi*ts* or bits, where each bit is either a 0 or a 1.

An example of a PCM system is called the T1 carrier system (Fultz and Penick, 1965). It was pioneered by the Bell System in the United States in the early 1960s and has been adopted for use throughout the United States, Canada and Japan.

In the T1 carrier system, a speech signal (male or female) is first low pass filtered at 3400 Hz because frequencies above this do not contribute to the perceived quality of the speech. The filtered signal is then sampled at 8 kHz, which is the *standard* sampling rate in telephone systems. The sampled signal is then compressed using the μ-law compressor and then quantized into 256 quantization levels which can

be encoded into 8 bits. Thus, at 8 kHz sampling rate, PCM gives a data rate of 64 kbps over the telephone network. The quality of the μ-law system is almost indistinguishable from the original continuous unquantized signal.

2.2.9 Differential Pulse Code Modulation (DPCM) analysis

When a speech signal is sampled at a rate higher than the Nyquist rate (Brigham, 1974), there is a high correlation between adjacent samples. This implies that the signal changes gradually from one sample to the next. Therefore, the difference between adjacent samples would have a smaller variance than that of the sampled signal itself.

In the standard PCM system as described in §2.2.8, the highly correlated samples are encoded directly. Because of the high correlation of the encoded samples, the encoded signal contains *redundant information.* By removing this redundancy, a more efficient coding scheme can be obtained.

The linear predictive analysis is one such technique and has been discussed in §2.2.5. It is based on the premise that if the behaviour of a speech signal is known up to a certain point in time, it is possible to make some *prediction* about its future values, based on the known past samples. Further, it is reasonable to conclude that the prediction would be more accurate for a signal with a smaller variance than one with a smaller one since a smaller variance implies that a signal varies less and therefore more predictable.

The Differential Pulse Code Modulation (DPCM) is a variation of the standard PCM technique. The difference between the PCM and DPCM is that in PCM, the speech samples are quantized directly (see §2.2.8) whereas in DPCM, it is the difference between the adjacent samples that is quantized.

The block diagram of a classic DPCM system is shown in Figure 2.15. In Figure 2.15, it can be seen that the DPCM consists of a quantizer, an encoder, and a linear predictive analyser. It should be noted that this is a special case of the general linear predictive analyser which has already been outlined in §2.2.5. The analyser in this case is a first order analyser, which means $p = 1$ in Equation (2.6).

Referring to Figure 2.15, it can be seen that the input to the quantizer is

$$e(n) = s(n) - \hat{s}(n) \tag{2.11}$$

which is the difference between the unquantized input sample, $s(n)$, and its predicted value, denoted by $\hat{s}(n)$. Note that the predicted value is the output of the first order linear predictive analyser.

If we denote the output of the quantizer and the quantization error (see Figure 2.13) by $e_q(n)$ and $\delta(n)$, respectively, then the output of the quantizer may be expressed as

$$e_q(n) = e(n) + \delta(n) \tag{2.12}$$

From Figure 2.15, one can also see that the input of the linear predictive analyser is derived from the sum of the predicted sample, $\hat{s}(n)$, and the output of the quantizer, $e_q(n)$.

$$s_q(n) = \hat{s}(n) + e_q(n) \tag{2.13}$$

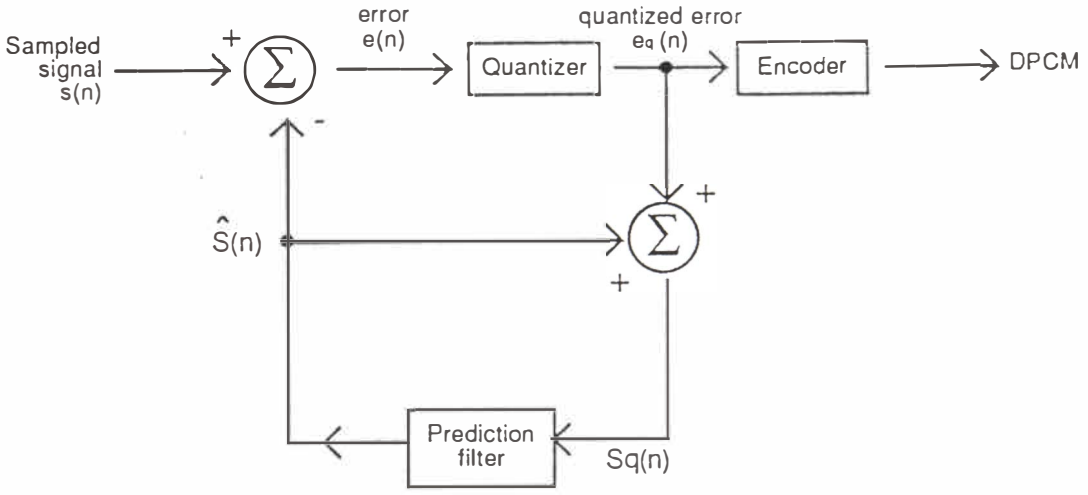


Figure 2.15: *Block diagram of a classic DPCM system.*

By substituting Equation (2.12) into Equation (2.13), the following equation results:

$$s_q(n) = \hat{s}(n) + e(n) + \delta(n) \tag{2.14}$$

Notice that Equation (2.14) may be rewritten as

$$s_q(n) = s(n) + \delta(n) \tag{2.15}$$

by replacing the first two terms in the right hand side of Equation (2.14) with $s(n)$.

Equation (2.15) means that the quantized version, $s_q(n)$, differs from the input signal, s_n by the quantization error, $\delta(n)$, irrespective of the properties of the linear predictive analyser. Therefore, if the prediction is good, the variance of the prediction error $e(n)$ will be smaller than the variance of $s(n)$, so that a quantizer with a given number of levels can be adjusted to produce a quantizing error with a smaller variance than would be possible if the input signal $s(n)$ were quantized directly as in PCM system.

The DPCM system has been shown to produce speech quality of the same quality as the PCM system but at a bit rate of 32 kbps, as opposed to 64 kbps of the PCM system (Schroeder, 1984; Andrews, 1984).

2.2.10 Adaptive DPCM analysis

The discussion on PCM brings out the two conflicting requirements in quantizing speech signal. Firstly, the quantization step size must be large enough to accommo-

date the maximum peak-to-peak range of the speech signal while maintaining the lowest possible number of quantization levels. Secondly, the quantizing step size must be small enough to minimize the variance of the quantization error (or noise). This is further complicated by the nonstationary nature of the speech signal. The amplitude of the speech signal can vary over a wide range, depending of the speaker, the communication environment, and within a given utterance, from *voiced* to *unvoiced sounds* (see §1.4).

One approach which accommodates these requirements is by using a non-uniform quantizer. Two examples of non-uniform quantizers are the μ-law and the A-law quantizers and have already been discussed in §2.2.8. A second approach is to quantize the difference of the speech signal rather than the speech signal itself. The second approach is called the DPCM and has been examined in §2.2.9.

The following paragraphs will explain the principles of the third approach, which is known as the Adaptive Differential Pulse Code Modulation (ADPCM). As its name implies, the quantizing step is adjusted automatically to match the variance of the input speech signal. This is achieved by varying (for every new input sample, in general) the step size of the quantizer, based on the knowledge of the quantizing step used for the previous samples (Jayant, 1973).

In its simplest form, the ADPCM operates with a one-word memory. Consider a n-bit uniform quantizer as an example. This means the output of the quantizer may assume any one of the 2^n allowable levels as shown in Equation (2.16).

$$y_i = H_i\delta_i \qquad H_i - \frac{\delta_i}{2} < x_i < H_i + \frac{\delta_i}{2} \tag{2.16}$$

where
$H_i = 1, 2, 3, ..., 2^n$ and δ_i is the step size.

The next step size δ_{i+1} is now chosen to be the previous step size multiplied by a function of the present quantized level, H_i, as indicated by

$$\delta_i = \delta_i . f(H_i) \tag{2.17}$$

When the multiplier function $f(H_i)$ is designed properly, the adaption rule defined by Equation (2.16) and Equation (2.17) serve to match the step size to an updated estimate of the variance of the input signal.

In ADPCM, both the quantizer and linear predictive analyser are adaptive. The adaptive nature of the quantizer has been explained in the last paragraph. In adapting the predictive analyser, it is common to assume that the statistical properties of the speech signal remain fixed over relatively short time intervals (say 20ms). The coefficients of the linear predictive analyser are then calculated to minimize the average squared prediction error over the 20ms interval and then updated for every 20ms segment of speech thereafter.

In a study of ADPCM, it has been shown (Cummiskey *et al.*, 1973) that listeners preferred speech coded using ADPCM to that using standard PCM with logarithmic compression at the same bit rate. It is now generally accepted (Schroeder, 1984) that ADPCM at 32 kbps can produce speech of the same quality as standard PCM at 64 kbps.

Figure 2.16: *Schematics of a typical pitch analyser.*

2.2.11 Pitch analysis

An estimate of the pitch or fundamental frequency of a speech waveform is often crucial to the performance of speech-based systems. The number of pitch analysis algorithms which have been reported (Hess, 1983) is a testimony to the importance of the pitch information in speech processing research.

Accurate pitch control is important to the intelligibility and naturalness of synthesised speech. The pitch of a speech waveform also conveys much of the prosodic information in the speech, for example, the pitch will rise towards the end of a question. This information, when displayed in real time, has been found to be a very useful tool for speech therapists (Watson *et al.*, 1988). Further, Laver *et al.* (1984) have found that the pitch contains information regarding the medical condition of a speaker's vocal cords and thus offers a potential diagnostic screening mechanisms.

Figure 2.16 shows the block diagram of a typical pitch analyser. As shown in Figure 2.16, a typical pitch analyser consists of a preprocessor, a pitch analysis algorithm, and a post-processor. The preprocessor serves to condition the input speech signal so that it is more suitable for the pitch analysis algorithm. This is achieved by either 1) reducing the effects of the formant frequencies by using a low pass filter or 2) by enhancing the effects of the fundamental, such as centre clipping. The postprocessor is used for smoothing the output of the algorithm and for correcting any gross errors made by the algorithm. Gold and Rabiner (1969) have found that the median filter was particularly useful for correcting spurious errors made by even the best pitch analysis algorithms.

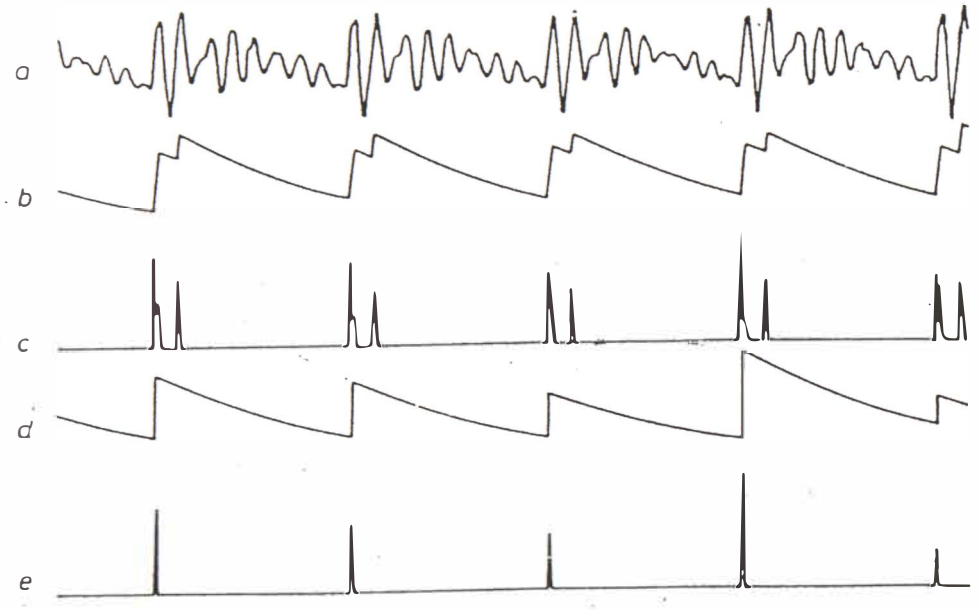


Figure 2.17: *The pitch analysis algorithm of Gruenz and Scott (1949). (a) The original speech waveform. (b) The waveform derived from (a) by following the upward movements of the speech signal until a peak is encountered and then decays exponentially until it crosses the speech signal again. (c) The pulse train obtained by differentiating the waveform shown in (b). (d) The result of performing step (b) on the waveform shown in (c). (e) The result of differentiating the waveform shown in (d). The pitch is estimated from the time intervals between the pulses shown here.*

Pitch analysis algorithms may operate in the time domain or the frequency domain. In the following paragraphs, a discussion on the principles of the pitch analysis algorithms in these two domains are presented.

The simplest time-domain algorithm consists of a low-pass filter followed by a zero-crossing detector. Unfortunately, this algorithm fails when the fundamental frequency is near the formant frequencies. This is often the case for female voices (since the female voices are higher pitch) and for many male voices articulating high-pitched /i/ or /u/ vowels. This problem may be alleviated by employing a threshold crossing instead of a zero crossing detector. In using a threshold detector, it is advisable to use one which adjust the threshold automatically so that a wide range of the amplitude of the signal can be catered for.

Apart from the zero crossing, one other feature which is used for pitch analysis is the peak of the speech waveform. One such scheme, invented by Gruenz and Scott (1949), is depicted in Figure 2.17. In Figure 2.17a, the original waveform is shown. From Figure 2.17a, a new waveform (Figure 2.17b) is obtained by following the original speech waveform until a peak is reached. Once a peak has been found, it

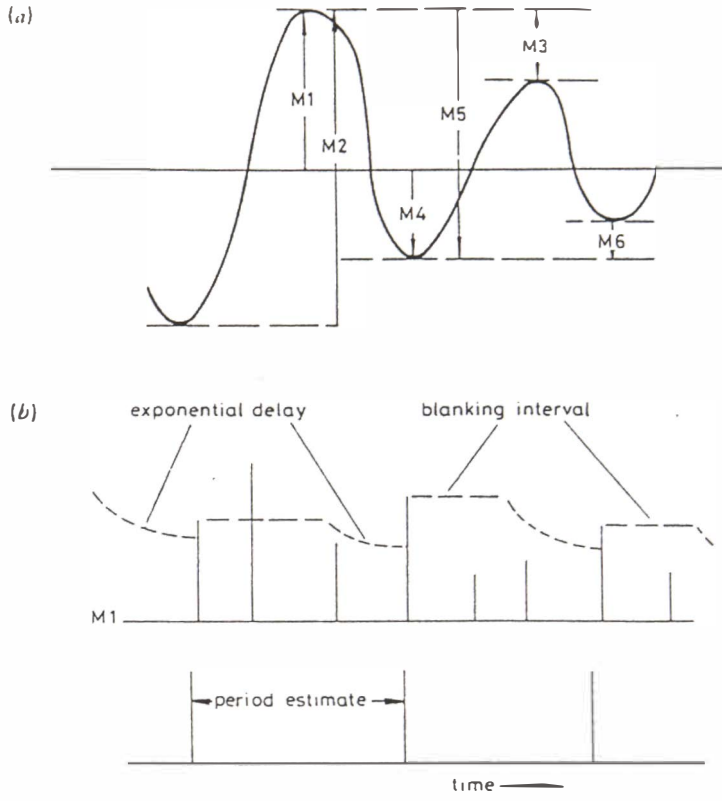


Figure 2.18: *(a) The period markers used in the pitch analysis algorithm of Gold and Rabiner (1969). (b) The periods between each of the markers shown in (a) are estimated by a blanking interval, during which the markers are ignored, followed by an exponential decay until the next marker is encountered.*

then decays exponentially until it crosses the speech signal again. Figure 2.17c is produced by differentiating the resulting waveform. It can be seen from Figure 2.17c that the pulses at the beginning of each glottal cycle has been picked up, along with some erroneous pulses. The processes indicated by Figure 2.17b and Figure 2.17c are then repeated, with the resulting waveforms as shown in Figure 2.17d and Figure 2.17e. Figure 2.17e shows that the start of each glottal cycle (indicated by the regularly spaced pulses) has been correctly identified. The pitch is then estimated from the time intervals between these pulses.

Gold and Rabiner (1969) implemented a more elaborate time domain pitch analyser. They extracted six parameters ($M1 - M6$) from the speech waveform to be analysed. These six parameters, as shown in Figure 2.18a are used as markers for the pitch periods. From each of these six markers ($M1 - M6$), estimates of the periods can be obtained. Figure 2.18b shows the way in which this is achieved, where $M1$ is used as an example. Figure 2.18b shows the time waveform of the period marker $M1$. This waveform is then analysed by a peak detector, followed by a blanking interval and an exponential decay, as shown in Figure 2.18b. This removes the erroneous period markers in exactly the same way as the pitch analysis algorithm (Gruenz and Scott, 1949) which was described earlier. The pitch periods are then estimated from the remaining markers.

The process is repeated for the other five markers, $M2-M6$. This gives rise to five

corresponding estimates of the pitch periods. From these estimates, the best estimate is decided upon by an elaborate procedure detailed in Gold and Rabiner (1969).

Tucker and Bates (1978) devised a similar pitch analysis algorithm. Unlike the method used by Gold and Rabiner (1969), Tucker and Bates (1978) used an adaptive centre clipper to remove the minor maxima and minima of the speech wave. This reduced the number of peaks and hence the amount of computation as well. The features that were used were also different. The features that were used by Tucker and Bates (1978) were the amplitude, energy, polarity and a shape factor of each peak as well as the intervals between peaks. This algorithm was designed for estimating the pitch of musical sounds and it is claimed that it operates over a range of 40-2400 Hz. This is much more than is required for speech processing. More recently, Sutherland *et al.* (1988) examined and evaluated several versions of time domain pitch period (of speech) estimation algorithms. After the evaluation, they proposed an improved, multifeature algorithm, which is claimed to have better long term accuracy and more stable with respect to variation of the time origin of the input waveform.

The pitch of a speech waveform can also be obtained from the autocorrelation function of the speech waveform. The autocorrelation function of a digitised speech signal, $s(n)$, is given by

$$R(k) = \sum_{n=1}^{N} s(n)s(n+k) \tag{2.18}$$

where k, n, and M are integers. In Equation (2.18), M is the total number of samples in the waveform, and k is known as the delay. In other words, the autocorrelation function is obtained by multiplying the speech signal by a delayed version of itself. For a periodic signal the autocorrelation signal will exhibit peaks at delay values which are equal to the multiples of this periodicity. Thus, from the intervals between these peaks, the pitch can be calculated. In practise, it has been found (Rabiner, 1977) that better results can be obtained by multiplying the speech waveform with a smoothing function such as a Hamming window before the autocorrelation function is computed. Rabiner (1977) also demonstrated that if the signal is first preprocessed by centre-clipping, thus reducing the effects of the formants, the performance of the algorithm is improved.

The pitch estimation algorithms described thus far all operate by searching for periodicity in the time waveform. The pitch may also be estimated by first transforming the speech signal into the frequency domain.

If the speech waveform is periodic, then its spectrum will consist of individual lines separated by a distance which is equal to the reciprocal of the pitch period. However, as speech is only pseudo-periodic, its spectrum will be continuous with ripples at the fundamental frequency at its harmonics (*i.e.* frequencies which are of integer multiples of the fundamental). The fundamental frequency can be determined from the spacing of the harmonics if they are sufficiently visible.

A more objective technique for estimating the fundamental frequency from the spectrum is called cepstral processing (Noll, 1964). The first step of the technique is to compute the spectrum of the speech signal by Fourier transformation. The next step is to take the logarithm of the spectrum. Finally, the inverse Fourier transform of the logarithm of the spectrum is obtained (see §2.2.3).

The end result is called the cepstrum which has the dimension of time. Figure 2.7c shows a plot of a typical cepstral waveform. As can be seen from Figure 2.7c, the cepstrum has a number of narrow (short duration) peaks near the origin. These correspond to the impulse response of the vocal tract. Further away from the origin is a broader peak labelled as T_0. The location of T_0 can be used as an estimate of the pitch period. See §2.2.3 for more detailed description of this technique.

Another technique for estimating the fundamental frequency from the spectrum was reported by Schroeder (1968a). From the spectrum, he determined the frequencies of the higher harmonics and computed the lowest common divider of these frequencies to give an estimate of the fundamental frequency. Schroeder (1968a) claimed that his technique gave more accurate estimates than the cepstral technique.

2.3 History of speech recognition

The first attempt to build machines which could recognise speech was made around the 1940s (Ainsworth, 1988). In those days, to make a telephone call, a caller spoke to the operator at the exchange and gave the number of the person he/she wished to contact. The connection was then established by the operator. The engineers realised that, if a machine which could recognise spoken digits could be built, the operator could be dispensed with, and a more efficient and less expensive telephone system would result. However, these efforts failed because of their inabilities to cope with a variety of different voices. The alternative solution, the telephone dialling system was then introduced, and this has remained in use, with little change in principle, until today.

In 1950s, digital computers began to find applications in a variety of tasks. The media of communication between human beings and the computers were keyboards, printers and visual display units (VDU). The argument that speech, being the natural mode of communication between humans, should also be used in human-computer communication, was advanced. This argument provided a new motivation for research in speech recognition.

Bezdel and Chandler (1965) experimented with the recognition of five isolated vowel sounds, using only the zero-crossing information. Figure 2.19 shows the schematic diagram of the system. It consisted of a digital computer, a magnetic tape recorder, a device for extracting zero crossings, and another for converting the zero crossings into a form suitable for computer analysis. The method of extracting and measuring zero crossings is shown diagrammatically in Figure 2.20. The speech waveform, as well as the true zero level and the threshold levels are depicted in Figure 2.20a. A zero-crossing is said to have occurred when the speech waveform crosses the threshold levels. The zero-crossing locations of the speech waveform in Figure 2.20a, is indicated in Figure 2.20b. The threshold levels are used instead of the true zero level because they provide a means of eliminating noise during the absence of a signal. These zero-crossing locations are then represented by a rectangular wave (see Figure 2.20c) where the width of the pulses correspond to the distance between adjacent zero-crossings. A positive pulse means that the first crossing occurs

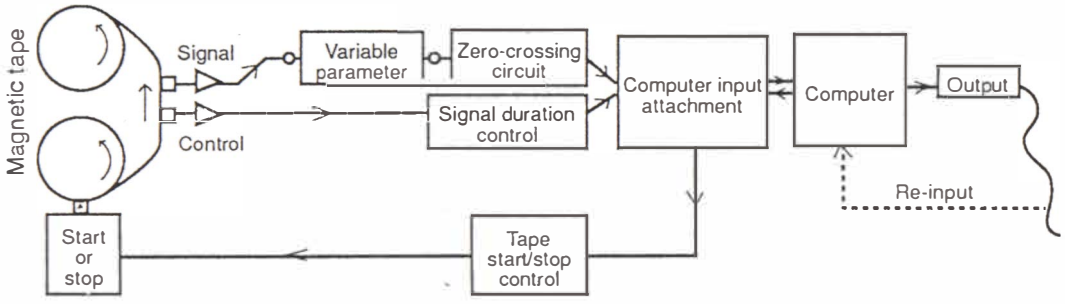


Figure 2.19: *The schematic diagram of the vowel recogniser of Bezdel and Chandler (1965).*

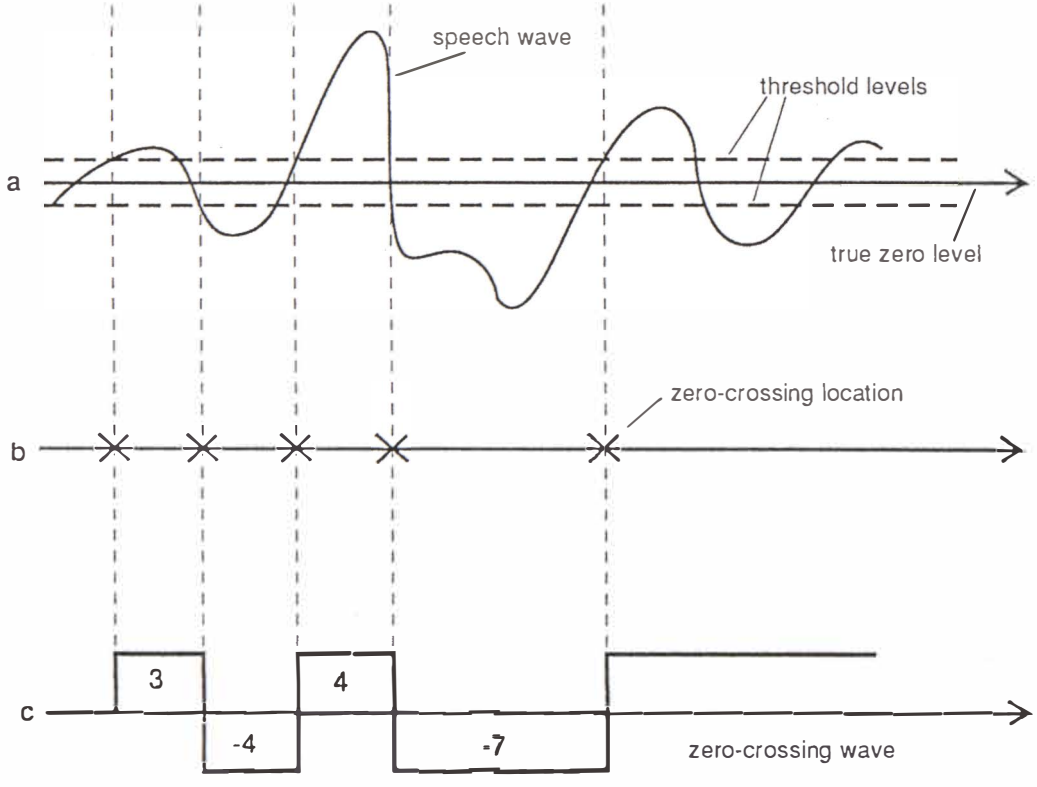


Figure 2.20: *Extracting zero crossing from a speech waveform.(a) speech wave, (b) zero-crossing location, (c) zero-crossing wave.*

at the positive threshold and the following crossing occurs at the negative threshold and vice versa. The area under the rectangular wave are then stored as numbers representing the distances between the crossings. These strings of numbers are called the zero-crossing distances.

These zero-crossing distances are then sorted out into N categories, each of which comprises a range of distances, for example, the first category may consists of zero-crossings which are of 0.1-0.2ms long while the second category may consists of zero-crossings which are of 0.2-0.3ms long and so on. The distribution, or the number of zero-crossings within each categories is then used as a parametric representation of the speech waveform of a particular vowel sound. Thus, by analysing a number of utterances from the same vowel, a reference distribution, X_r, for each vowel is obtained. When an unknown vowel is to be recognised, its waveform is analysed exactly as before. This results in an unknown distribution X and the Euclidean distance between X and each of the reference distribution X_r is computed. The reference distribution with the shortest distance is then chosen as the recognised vowel.

Bezdel and Chandler (1965) claimed that they obtained an average of 96% recognition rate of the five vowels spoken by the same speakers as those used to derive the reference data. In the case where the unknown vowels were spoken by a different group of speakers, the average recognition rate was found to decrease to 79%.

Unlike Bezdel and Chandler (1965) who uses the zero crossing rate directly for isolated vowel recognition, Ewing and Taylor (1969) uses information derived from the zero crossing rate of the speech waveform. It has been shown (Chang *et al.*, 1951) that the average rate of zero crossing of the undifferentiated speech wave is very nearly a measure of the first formant frequency. Furthermore, the average rate of zero crossing of the differentiated wave is a measure of the second formant frequency. Using this information, Ewing and Taylor (1969) calculated the first and second formants of the digits zero to nine and then computed the difference between these two formants. These difference signals are then computed for each of the digits and stored as references. To recognise an 'unknown' digit, its difference signal is first determined. After that, the Euclidean distances between the difference signals of the reference digits and the 'unknown' digit are calculated. The 'unknown' digit is then recognised as the reference digit with the smallest distance. Ewing and Taylor (1969) experimented with five male speakers and found that results were 'excellent' for recognising digits uttered by the same speaker but were not 'consistently successful' for recognising digits uttered by a different speaker.

During the 1970s and the early part of 1980s, work on speech recognition increased tremendously. Not only did the activity in research laboratories accelerated, but also the number of commercial speech recognisers increased considerably (Wallich, 1987). The use of dynamic programming algorithms has contributed to the advances made during this period.

A successful application of the dynamic programming algorithm was demonstrated by Itakura (1975). In this system, the LPCs of each of the word to be recognised are stored as reference templates. A similar set of the LPC pattern from an unknown word is then compared to the stored reference templates. However, be-

cause of the different speed at which human speaks, a meaningful comparison cannot be made unless these LPC patterns are time-aligned in some way. This is achieved by a dynamic programming algorithm (see Chapter 4) which non-linearly warps the time scale of one of the pattern until both the patterns are of the same length. The system was implemented on a DDP-516 computer and was capable of recognising 200 isolated words, spoken by a designated speaker with 97.3% accuracy. Sakoe and Chiba (1978) optimised a number of dynamic programming algorithms and achieved an impressive 99.8% recognition rate for the Japanese digits with their best systems.

While the dynamic programming technique was being explored extensively, Jelinik (1976) introduced the hidden Markov model technique (HMM) for speech recognition. Unlike the DTW which explicitly aligned two time waveforms (which may be a time sequence of LPCs), HMM technique does not require explicit time alignment. Instead, a probabilistic transition and observation structure is defined for each reference word. The HMM for each word includes (1) a state transition matrix probability matrix, (2) an initial probability vector, and (3) an observation probability matrix for discrete probability densities or a set of continuous densities defined by parameter sets, or a mixture of the two when different types of densities are used. During recognition, one computes for each given reference model the probability of observing the unknown sequence. The model that produces the maximum observation probability is chosen as the recognised word.

A typical speech recognition system based on the HMM was built by Levinson *et al.* (1983a). This system was trained on 1000 digits, spoken by 50 men and 50 women. A recognition rate of 96% was obtained. By 1984, several systems which are based on the HMM have appeared on the market, including the Verbex 1800 and the Dragon system.

A survey of the isolated word speech recognition systems was conducted by Rabiner and Wilpon (1987). From this survey, it can be seen that the performance of the isolated word speech recognition system has steadily improved over the years. In particular, the average error rate for the digits was 2.5% in 1979, and this has reduced to 0.4% in 1987. This represents a six-fold improvement in the error rates. This improvement is due to the fact that we have learnt more about the significant events in speech, and how to capture these events using appropriate analysis procedures and training algorithms.

More recently, a speech recognition system designed for use in moving vehicles has been marketed by a Swedish company (Blomberg *et al.*, 1987). It is a speaker-dependent, pattern matching word recognition system. The algorithms have been evaluated in moving cars. Noise compensation was achieved by adding the measured noise in the moving car to the reference patterns recorded in a parked car and by using a close-talking microphone. This improved the recognition rate from 69% to 97% on a ten-word vocabulary. They also conducted a more extensive tests with two cars and twelve speakers. The twenty word vocabulary contained some confusable words and was trained in a parked car. During 98 sessions, 1960 words were read under different conditions with an average recognition rate of 86%. With closed windows at 90 km/h, the mean was 91%. An open window at the same speed decreased the result to 82%.

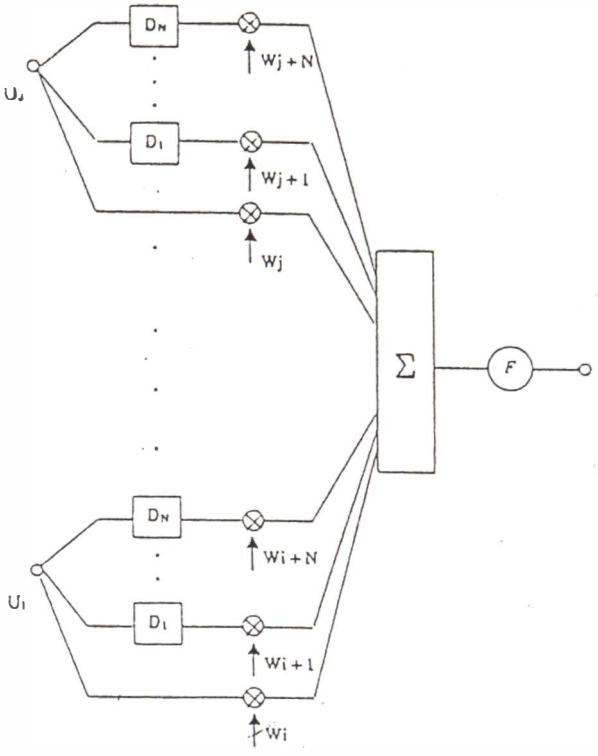


Figure 2.21: *A Time-Delay Neural Network (TDNN) unit.*

Apart from the DTW and HMM techniques, neural networks techniques (see Chapter 4) have also been used frequently for speech recognition (Lippmann, 1987; Kohonen *et al* 1984; Kohonen, 1988; Waibel *et. al.* 1989).

Two examples of speech recognition systems that uses neural network techniques will be described here. The first example is a Time Delay Neural Networks (TDNN) designed by Waibel *et al.* (1989). Figure 2.21 shows a block diagram of the basic TDNN unit. As shown in Figure 2.21, each TDNN unit has J inputs and each input is then passed through delays D_1 to D_N. All the J inputs and their delayed versions are then weighted and added together. For $N = 2$ and $J = 16$, for example, 48 weights will be needed to compute the weighted sum of the 16 inputs, with each input now measured at three different points in time. In this way, a TDNN unit has the ability to relate and compare current input to the past history of events. These weighted sum of the inputs are then passed through a non-linear function F. The non-linear function used in this case is the sigmoid function $f(\alpha)$:

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha-\theta)}} \tag{2.19}$$

where α is the input, e is the exponential function, θ is the internal threshold or offset. Figure 2.22 shows a plot of the sigmoid function with zero offset, *i.e.* $\theta = 0$.

The TDNN recognition system constructed by Waibel *et. al.*(1989) consists of one input layer, two hidden layers and an output layer of the basic units shown in Figure 2.21. The arrangement of these TDNN units into layers allows for the formation of arbitrary nonlinear decision surfaces. One important feature of Waibel's

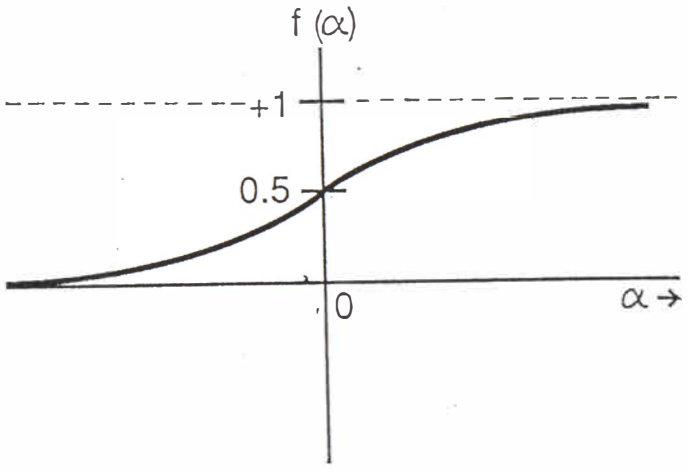


Figure 2.22: *The graph of sigmoid function (see text) with* $\theta = 0$.

system is that these decision surfaces were computed automatically using error-backpropagation (Lippmann, 1987). Furthermore, the time-delay arrangement enables the network to discover acoustic-phonetic features and the temporal relationships between them independent of position in time and hence not blurred by temporal shifts in the input (Waibel *et al.*, 1989).

As a recognition task, the speaker-dependent recognition of the phonemes "B", "D" and "G" in varying phonetic contexts was chosen. The recognition rate obtained from over 1946 testing tokens from three speakers was 98.5%. Waibel *et al.* (1989) claimed that the TDNN also managed to abstract some acoustic-phonetic features such as the onset of vowels and the rise and fall of the second formants.

The second example of neural network speech recognition system was designed by Kohonen (1988). Unlike most systems where the basic recognition unit is a complete word, Kohonen's system uses the *phonemes* (distinct sounds which makes up a word) as the basic recognition unit. Because it uses the phonemes as the basic recognition units, the system can theoretically transcribes dictation from an unlimited vocabulary. The system has been implemented in hardware and incorporated in an IBM-PC. It has been reported that the time taken by the system to adapt to a new speaker is less than 10 minutes. The recognition rate achieved was between 96 to 98% with a 1000-word vocabulary. The system can also achieve near real time recognition for unlimited text as well as isolated words: the mean delay being of the order of 250 ms per word.

2.4 History of speech coding

Throughout the ages, many speech coding schemes have been proposed. The PCM, ADPCM, subband coding, linear predictive coding, multi-pulse linear predictive coding and stochastically excited linear predictive coding are examples of these. The first three are collectively known as waveform coders while the last three are often referred to as source coders.

The purpose of waveform coders, as their names imply, is to reproduce, as faithfully as possible, any signal waveform without any assumption about the origin of the signal (Flanagan *et al.*, 1979). Because no assumptions about the origin of the signal are made, they perform equally well with a wide range of signals – speech, music, tones and voiceband data. They also tend to be robust for a wide range of speaker characteristics and for noisy environments.

By adapting the waveform coders to a specific type of signal, greater efficiency can be achieved. In speech applications, this is attained by observing the statistics of the speech signal, so that the encoding error is minimized.

In contrast to waveform coders, source coders assume *a prior* knowledge about how the signal to be coded is generated at the source. The rationale is that certain physical constraints of the signal generation can be quantified, and then turned to advantage in coding the signal efficiently. The implies that the signal must be fitted into a specific signal generation model which can be characterised by parameters.

In speech coding, the conventional speech generation model is the source-filter model (see §1.3). The parameters of this model are the pitch of the speech signal to be coded, whether the speech signal is voiced or unvoiced and the impulse response of the vocal tract filter. By careful adjustment of these parameters, very high quality coded speech can be achieved.

In the following sections, the development of some examples of waveform coders and source coders are described.

2.4.1 Waveform coders

2.4.1.1 PCM

PCM is a classic example of a waveform coder. Its development as a speech analysis technique has been discussed in §2.2. PCM technique and its derivatives (log-PCM, DPCM and ADPCM) have been adopted as the CCITT standards for speech coding in telephone links (Jayant, 1986) with very good results. The standard PCM coding scheme (see Figure 2.12) yields highly acceptable quality at 64 kbps; in fact, few people can tell whether the voice at the other end of the telephone line has been transmitted digitally.

2.4.1.2 LOG-PCM

The success of the standard PCM scheme has encouraged further studies into more efficient speech coding techniques. These studies have shown that in speech waveforms, the low amplitudes occur more frequently than the high amplitudes. Furthermore,

Licklider (1960) established that for voiced speech, which has large amplitudes, the intelligibility remains very high even when the amplitudes are quantized to 1 bit, *i.e.* only the signs of the amplitudes are transmitted. These two findings have led to the development of the μ-law (Smith, 1957) PCM and the A-law (Kaneko, 1970) PCM coding schemes. The difference between these two modified schemes and the standard PCM is that the modified schemes employ non-uniform quantization where the fineness of the quantization is inversely proportional to the amplitude of the input speech. This means that low amplitudes, which predominate in speech, are quantized with imperceptable quantization error whereas large amplitudes are more coarsely quantized. The implications of Licklider's findings are that the higher quantization errors in the higher amplitudes are masked (to some extent) by the large amplitudes and are not perceived by human hearing. These two modified PCM coding schemes, which yields highly acceptable quality at 64 kbps, are sometimes collectively known as the log-PCM schemes and have replaced PCM as the standard.

2.4.1.3 DPCM/ADPCM

In order to further reduce the bit rate yet maintain a satisfactory speech quality, new coding strategies are needed. These new strategies must be able to eliminate the redundancies in the speech signals more completely. In addition, the nonredundant parts of the signals must be encoded in such a way that the degradation caused by the resultant quantization errors is not perceived by human. Conversely, if there are two schemes operating at the same bit rate, the one with the more efficient (in the sense that the perceived degradation is less) allocation of bits will give higher quality speech. This claim is vindicated, as we have seen, by the fact that log-PCM has replaced PCM as the standard coding schemes at 64 kbps.

The next stage of development in speech coding is the DPCM and ADPCM. Like log-PCM, DPCM and ADPCM are derived from the standard PCM scheme. They operate on the same principles as the PCM. However, both DPCM and ADPCM encode the differences between successive samples rather than the samples as in PCM. This is because successive samples of speech are highly correlated. Therefore, by taking the difference between successive samples, the resultant signal will occupy a smaller range than the original signal (Haykin, 1983). This means that DPCM and ADPCM require less quantization levels (or bits) in order to achieve the same quantization precision as the PCM or log-PCM schemes. By adaptively changing the quantization step size, ADPCM can achieve even better quality speech than DPCM. It has been found the quality of ADPCM coded speech at 32 kbps is the same as PCM coded speech at 64 kbps. Further details about DPCM and ADPCM can be found in §2.2.

2.4.1.4 Subband coding

Subband coding of speech is a relatively mature form of waveform coding of speech. The speech is first subdivided into a number of subbands which are then individually encoded. The underlying principle for the coder is that bit allocation can be weighted so that those subbands with the most important information get the most

bits. The initial subband coders used fixed bit allocation based on the average spectrum of speech. The coder of Corchiere *et al.* (1982) is typical of this generation of coders. This was very quickly superseded by another subband coder introduced by Ramstad (1982). The main feature of the subband coder designed by Ramstad was that the bits were allocated adaptively.

More recently, a sub-band coder capable of producing 'good' quality speech at 16 kbps was designed by Cox *et al.* (1988). The coder was implemented on the AT&T DSP32 signal processing chip. The design was based on the work of Ramstad. The speech is first of all divided into six subbands, each of 500 Hz wide. The signal from each subband is then blocked into 16 ms frames and the rms value of each frame is quantized and transmitted. This quantized rms value takes up 2 kbps of the bit rate. Using an iterative procedure, the remaining 14 kbps are allocated adaptively to the subbands.

2.4.2 Source coders

2.4.2.1 Linear predictive coefficient (LPC) Vocoder

The classical source coder is the speech coder of Atal and Hanauer (1971). This speech encoder, which is based on linear predictive analysis, can be said to be the precursor of subsequent source coder techniques. The speech coder consists of two functional blocks: the analysis part and the synthesis part. During the analysis, 15 parameters are calculated from segments of speech which are either one pitch period (for voiced speech) or 10ms long (for unvoiced speech). These 15 parameters are the 12 LPCs, the pitch period, voiced/unvoiced parameter, and the rms value of the speech samples. These 15 parameters are encoded and transmitted to the synthesiser.

During the synthesis, the speech is reconstructed using these 15 parameters. A block diagram of the speech synthesiser of Atal and Hanauer (1971) is shown in Figure 2.23. As shown in Figure 2.23, the excitation source may be a pulse generator or a white-noise generator. The selection between the pulse generator and the white-noise generator is made by the voiced-unvoiced switch. Notice that while the white-noise generator produces uncorrelated uniformly distributed random samples with standard deviation of 1 at each sampling instant, the pulse generator produces a pulse of unit magnitude at the beginning of each pitch period. The amplitude of the excitation signal is adjusted by the amplifier G whose gain g is selected to provide the correct power in the synthesised speech signal. Thus,

$$\delta_n = g e_n \tag{2.20}$$

where e_n, δ_n are the nth sample of the original and the amplified versions of the excitation signal respectively. The linearly predicted value $\hat{s}_n$ of the speech signal is then combined with the amplified excitation signal δ_n to form the nth sample of the synthesised speech signal s_n, as shown in Equation (2.21):

$$s_n = \hat{s}_n + \delta_n \tag{2.21}$$

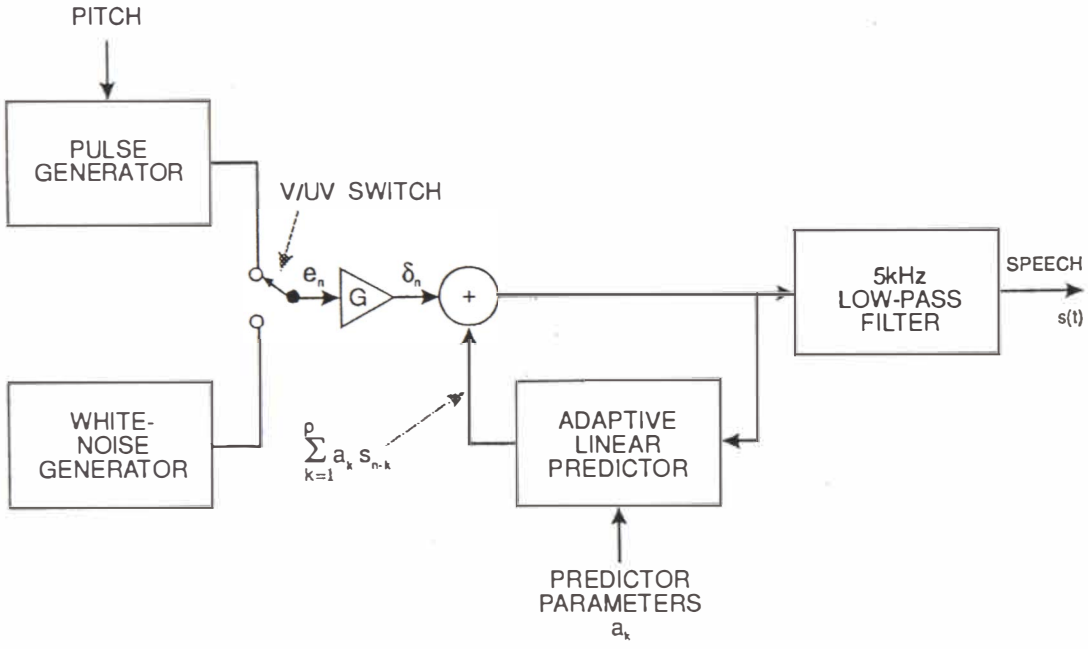


Figure 2.23: *Block diagram of the synthesiser of Atal and Hanauer (1971).*

where

$$\hat{s}_n = \sum_{k=1}^{12} a_k s_{n-k} \tag{2.22}$$

The speech samples s_n are finally low-pass filtered to provide the continuous speech wave $s(t)$ (see Figure 2.23).

As stated earlier, in Atal and Hanauer's system, 15 parameters are extracted from each frame of speech to be analysed. These 15 parameters are encoded using a total of 72 bits. The data rate in bits per second (bps) is obtained by multiplying the number of bits used to encode each frame of speech by the number of frames of speech stored or transmitted per second. For example, if 100 frames of speech is analysed per second, then this is equivalent to a bit rate of 7.2kbps. By reducing the frame rate to 33 per second, a bit rate of 2.4kbps can be achieved. It was reported that informal listening tests show very little or no perceptible degradation in the quality of the decoded speech, even at 2.4 kbps. The above bit rate is about 30 times lower than that of the standard PCM encoder which, at 8kHz sampling rate and coding with 8 bits/sample, gives a bit rate of 64kbps.

The success of the LPC vocoder described above has resulted in the development of numerous vocoders based on the same principle but with varying degree of modifications. The design of Wong and Markel (1977) is one example. Wong and Markel (1977) investigated two modifications. The first modification was to represent unvoiced speech with fewer LPCs. Their subjective tests indicated that as few as four coefficients could be used without noticeable effect. This observation was also

supported by the Diagnostic Rhyme Test (DRT) (IEEE, 1969). The second modification that Wong and Markel (1977) implemented was to double the frame rate for unvoiced speech from 44.4 frames/sec to 88.8 frames/sec. The second modification resulted in a significant improvement in the DRT scores.

2.4.2.2 LPC/VQ

Wong *et al.* (1982) developed an 800 bps LPC vocoder which used vector quantization (see Chapter 3) to encode the LPC coefficients. They claimed that the 800 bps vocoder preserved most of the intelligibility using 10 LPC coefficients. It was also robust under different transmission errors and acoustic conditions. They have also found that the speech quality of the vector quantization vocoder is acceptable and sometimes very close to LPC encoded speech at 2.4 kbps.

Bukiet *et al.* (1987) implemented an LPC/VQ encoder in hardware. The system used an INTEL 8751 micro-controller and two digital signal processors: the Texas Instruments TMS320c20 for data acquisition and the AT&T DSP32 for speech coding. The system is a portable unit which can perform LPC/VQ speech coding at a range of bit rate starting from 400 bps. The system requires 10 ms in order to analyse, encode and to resynthesise 20 ms of speech. Bukiet *et al.* (1987) reported comparable DRT scores with those published by Wong *et al.* (1982).

2.4.2.3 Multipulse linear predictive coding

In the classical LPC vocoders, speech is classified into only two categories – voiced and unvoiced. Further, speech is modelled as the output of an all-pole (linear predictive) filter, driven by an excitation function. In the case of voiced speech, the excitation consists of a train of pulses separated by the pitch period for voiced speech and pseudorandom noise for unvoiced speech. While the output of this method is highly intelligible, it is not natural sounding even at high bit rates (Singhal and Atal, 1989). While the cause of the unnaturalness are not well understood, it is believed that the main problem is in classifying speech as either voiced or unvoiced and modelling the corresponding excitation as pitch pulses or white noise. There are portions of speech where it is uncertain whether the speech is voiced or unvoiced. Moreover, accurate determination of pitch period in voiced speech is sometimes difficult (Singhal and Atal, 1989).

The problems mentioned in the last paragraph can be alleviated by using the multipulse excitation model, first proposed by Atal and Remde (1980). A simple multipulse LPC (MPLPC) encoder is shown in Figure 2.24. As shown in Figure 2.24, the excitation generator produces a sequence of pulses, $u(n)$, as the input to an LPC all-pole filter. The output, $\hat{s}(n)$, is then subtracted from the original speech, $s(n)$, to gives an error signal. The error signal is then suitably weighted and then fed back to the excitation generator. The excitation generator computes the amplitudes and locations of the pulses such that the weighted error (indicated by $e(n)$ in Figure 2.24) is minimised.

The excitation of the MPLPC model consists of a few pulses, regardless of whether the speech is voiced or unvoiced. Because the same form of excitation

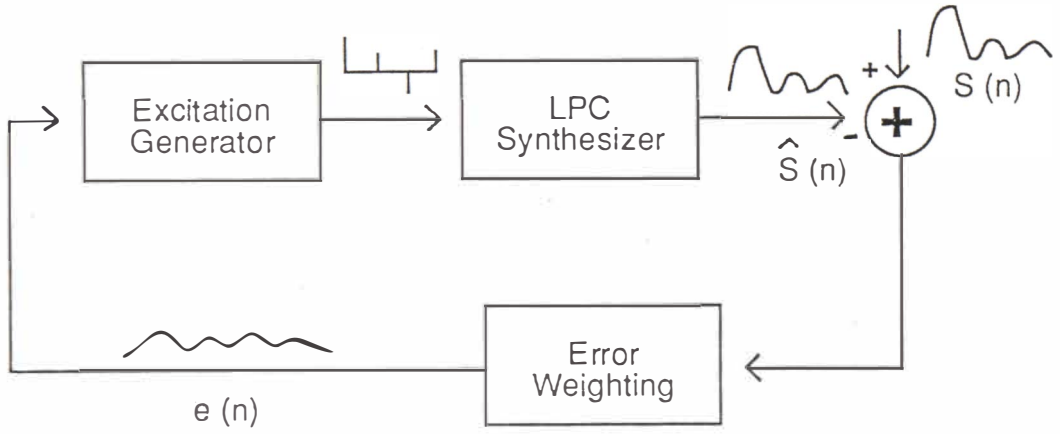


Figure 2.24: *Block diagram of a multipulse LPC encoder*

is used for all speech sounds, the problems of pitch estimation and voiced-unvoiced analysis are avoided. The number of pulses needed in each frame of speech depends on the desired speech quality, the more pulses per frame, the better the quality. However, as the positions and the amplitudes of the pulses have to be transmitted, a quality versus bit-rate trade-off must be made. In general, 6-8 pulses for every 10ms of speech is sufficient for high quality speech (Singhal and Atal, 1989).

Although the multipulse excitation model is conceptually simple, the optimization of the pulse amplitudes and locations is computationally intensive. In particular, if a speech segment of N samples is to be encoded with m pulses, there will be $\binom{N}{m} = N!/m!(N-m)!$ possible combinations of pulse locations. Thus, an exhaustive search for the global optimal solution is clearly impractical, especially as the number of pulses m increases. The required computation can be kept within reasonable bounds by searching for pulse locations in stages; at each stage, the amplitude and location of only one pulse is allowed to vary. This reduces the computation required to m searches of order N (Atal, 1985).

Many algorithms for computing the amplitudes and locations of the pulses can be found in the literature. The algorithms devised by Boyd (1987), Fukui and Shibagaki (1987), and Lefevre and Passien (1985) are examples. These are of varying degrees of optimization and computational complexity. Some of these have been implemented in hardware. In particular, Fukui and Shibagaki (1987) implemented a MPLPC with pitch prediction on a single-chip 32-bit floating-point signal processor (μPD 77230) and compared the performance with three other MPLPC algorithms. They found that the computation required by their method is 20% less than that

required by the other three methods while still achieving comparable quality of resynthesised speech at 8-9.6kbps.

2.5 Summary

A detailed account of the historical development of speech processing research has been provided. In particular, I have traced through chronologically, the development of computer speech synthesis, speech analysis, speech coding and speech recognition. Recent development in each area has also been included in the review. From these accounts, it can be seen that these developments are very closely linked and that the differences between them are very subtle and in fact very difficult to talk about each area in isolation. However, it is notable that these speech processing research areas involve more or less the same mathematical principle and they can be divided according to their specific applications.

Chapter 3

MATHEMATICAL PRELIMINARIES

"Speech is civilisation itself. The word, even the most contradictory word, preserves contact – it is silence which isolates."
(Thomas Mann, The Magic Mountain 1924)

This chapter deals with the mathematics of two very important techniques of speech processing: the linear prediction technique (Makhoul 1975; Markel and Gray 1976; Atal and Hanauer 1971; Thorpe 1990; Rabiner and Schafer 1978) and the vector quantization technique (Makhoul *et al.* 1985; Jayant and Noll 1984). This detailed treatment is needed for later chapters in the thesis, in order to appreciate the significance of the work reported there.

3.1 Linear prediction in speech processing

From the discussions in §2.1, §2.2.5, §2.4, it is evident that linear prediction has been widely used for speech synthesis, analysis, coding and other areas of speech processing. Unlike the discussions in the aforementioned sections where the emphasis is on historical development, the following discussions will emphasise some mathematical aspects of the linear prediction technique.

3.1.1 The generalized linear prediction model

The following mathematical expositions assume that the continuous-time speech signal, $s(t)$, under consideration is sampled to obtain a time-discrete signal, $s(nT)$, where n is an integer variable and T is the sampling interval. From here on, $s(nT)$, $s(n)$, s_n are used interchangeably to denote the time-discrete signal at the n^{th} sam-

pling instant. It should be mentioned that the the same "mathematics" applies whether or not the amplitudes of $s(nT)$ are discretized.

Consider a speech signal, s_n, which is regarded as the output of some unknown system with some unknown input, u_n, such that the following relation holds:

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G\sum_{l=0}^{q} b_l u_{n-l}, \quad b_0 = 1 \tag{3.1}$$

where the predictive coefficients $a_k, 1 \leq k \leq p, b_l, 1 \leq l \leq q$, and the gain G are the parameters of the unknown system. One interpretation of Equation (3.1) is that the output s_n is a linear function of past outputs and present and past inputs. That is, the signal s_n is *predictable* from *linear* combinations of past outputs and inputs. Hence the name *linear prediction.*

Taking the Z-transform on both sides of Equation (3.1), we have from Equation (3.1):

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3.2}$$

where

$$S(z) = \sum_{n=-\infty}^{\infty} s_n z^{-n} \tag{3.3}$$

is the Z-transform of s_n, and $U(z)$ is the z-transform of u_n. $H(z)$ in Equation (3.2) is known as the transfer function of the system. In Equation (3.2), the roots of the numerator and denominator polynomials are known as the zeroes and poles of the model, respectively. Correspondingly, $H(z)$ is known as the general *pole-zero* model.

From the general pole-zero model, two special cases of the pole-zero model can be derived:

1. all-zero model: $a_k = 0, \quad 1 \leq k \leq p$
2. all-pole model: $b_l = 0, \quad 1 \leq l \leq q$

Because the all-pole model is the most widely used model, the remaining discussions will be restricted to this model only.

3.1.2 The all-pole linear prediction model

The all-pole prediction model of speech signal is derived from Equation (3.1) by substituting $b_l = 0$ to give:

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G u_n \tag{3.4}$$

Similarly, the transfer function $H(z)$ in Equation (3.2) reduces to

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3.5}$$

The all-pole model is shown in Figure 3.1. It is clear, from Equation (3.4) or Equation (3.5), that if a_k and u_n are known, then successive samples of s_n can

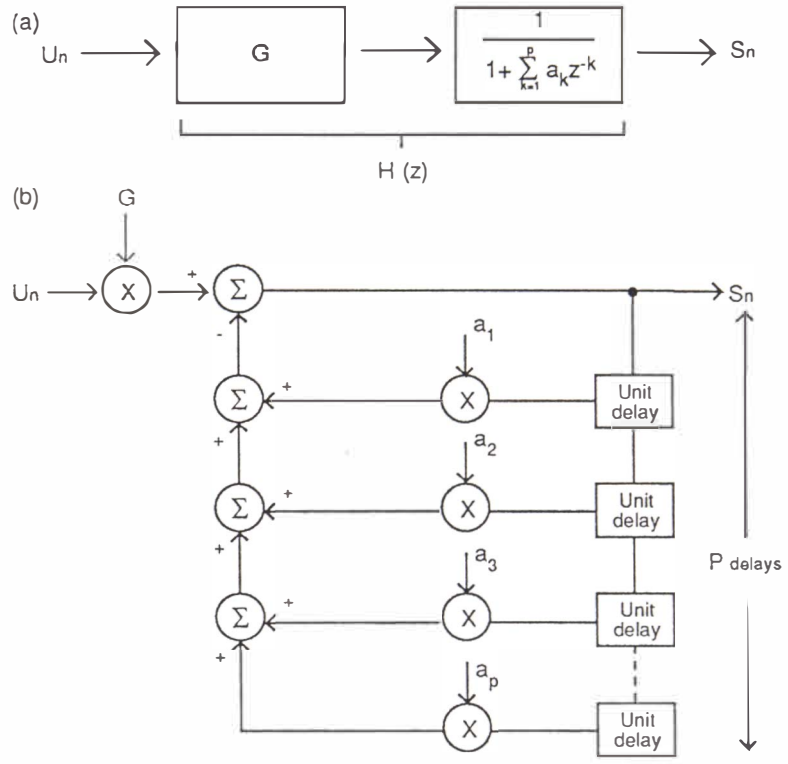


Figure 3.1: *All pole modelling of speech in (a) z-domain and (b) time domain.*

be computed exactly (as linear combination of past samples and past and present inputs). However, in the speech processing applications mentioned in §2.1, §2.2.5, §2.4, the speech samples s_n is usually the only information available directly to an experimenter and u_n is usually not available directly. Therefore, in many speech processing applications, the problem is to determine the predictor coefficients a_k and the gain G in some manner so that the speech signal s_n can be reconstructed later from the predictor coefficients and the gain.

3.1.3 Calculating predictor coefficients

It is first of all assumed here that the input u_n in Equation (3.4) is unknown; *i.e.* $u_n = 0$. Therefore, Equation (3.4) becomes

$$\hat{s}_n = -\sum_{k=1}^{p} a_k s_{n-k} \tag{3.6}$$

where $\hat{s}_n$ is the approximate estimate of the present sample s_n given in Equation (3.4). The error between the actual sample s_n and the predicted value $\hat{s}_n$ is thus given by:

$$e_n = s_n - \hat{s}_n = s_n + \sum_{k=1}^{p} a_k s_{n-k} \tag{3.7}$$

where e_n is also known in the literature as the *residual*. The procedure for determining an optimal set of predictor coefficients $a_k, k = 1, 2, ..., p$, which minimises the

total squared error e_n^2 is derived here. For simplicity, the range of summation of e_n^2 is not dealt with here but is deferred until §3.1.3.1.

Denoting the total squared error to be minimised by E, where

$$E = \sum_n e_n^2 = \sum_n \left\{ s_n + \sum_{k=1}^{p} a_k s_{n-k} \right\}^2 \tag{3.8}$$

E is minimized by setting the partial derivatives of E with respect to each of the a_k's simultaneously equal to zero. Hence,

$$\frac{\partial E}{\partial a_i} = \sum_{k=1}^{p} a_k \sum_n s_{n-i} s_{n-k} + \sum_n s_n s_{n-i} = 0, \qquad i = 1, 2, ...p \tag{3.9}$$

Notice that in Equation (3.9), the dummy variable for the partial differentiation is denoted by i, in order to avoid confusion with the a_k's in the summation.

Equation (3.9) can be rearranged to give:

$$\sum_{k=1}^{p} a_k \sum_n s_{n-k} s_{n-i} = -\sum_n s_n s_{n-i}, \qquad 1 \leq i \leq p. \tag{3.10}$$

which is a system of p linear equations with p unknowns *i.e.* all the a_ks. This system of equations is known as the *normal equations*. Since this system of equations is linear, it is possible to solve, by matrix inversion, all the predictor coefficients ($\{a_k,\ 1 \leq k \leq p\}$) which would minimize E in Equation (3.8) (Makhoul, 1975). By expanding Equation (3.8) and substituting Equation (3.10), the minimum total squared error, E_{min} can be shown (Makhoul, 1975) to be

$$E_{min} = \sum_n s_n^2 + \sum_{k=1}^{p} a_k \sum_n s_n s_{n-k} \tag{3.11}$$

3.1.3.1 Range of summation

The following paragraphs will deal with the range of the summation of e_n^2 which appear in Equation (3.8). Two ranges of summation will be considered. It is shown here that these two ranges of summation lead to two distinct methods for the estimation of the parameters.

3.1.3.2 Autocorrelation method

It is assumed here that the error E in Equation (3.8) is minimized over the infinite range $-\infty < n < \infty$. The autocorrelation function of the signal s_n is denoted:

$$R(i) = \sum_{n=-\infty}^{\infty} s_n s_{n+i} \tag{3.12}$$

Hence, Equation (3.10) and Equation (3.11) can be rewritten as

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i), \quad 1 \leq i \leq p \tag{3.13}$$

$$E_{min} = R(0) + \sum_{k=1}^{p} a_k R(k) \tag{3.14}$$

where $R(i)$ is as defined in Equation (3.12).

Equation (3.13) can be expanded in matrix form as

$$\begin{bmatrix} R_0 & R_1 & R_2 & \dots & R_{p-1} \\ R_1 & R_0 & R_1 & \dots & R_{p-2} \\ R_2 & R_1 & R_0 & \dots & R_{p-3} \\ \vdots & \vdots & \vdots & & \vdots \\ R_{p-1} & R_{p-2} & R_{p-3} & \dots & R_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_p \end{bmatrix} \tag{3.15}$$

Notice that the $p \times p$ autocorrelation matrix is symmetric (along the main diagonal) and the elements along any diagonal are identical. This type of matrix is known as the Toeplitz matrix (Grenander and Szegö, 1958). An elegant method devised by Durbin (1960) and Levinson (1947) exists for solving this special matrix. The Durbin-Levinson algorithm requires much less computational effort than other methods of solving symmetric equations (Witten, 1982).

Recall that the basic assumption of the autocorrelation matrix is that the range of summation is from negative infinity to positive infinity. In practice, we are quite often only interested in the signal s_n over only a finite interval, while at other times, the signal s_n is known over only a finite interval (Markel and A. H. Gray, 1973). This difficulty can be overcome by multiplying the signal s_n by a *window* function w_n to obtain another signal s'_n which is zero outside some interval $0 \leq n \leq N-1$:

$$s'_n = \begin{cases} s_n w_n, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

A commonly used window function is the Hamming window function (Thorpe, 1990). The autocorrelation function is then given by

$$R(i) = \sum_{n=0}^{N-1-i} s'_n s'_{n+i}, \qquad i \geq 0 \tag{3.17}$$

Figure 3.2 illustrates the effect of applying the Hamming window function to a segment of speech waveform 256ms long.

3.1.3.3 Covariance method

The covariance method is an alternative form of solution where the range of summation in Equation (3.8) is taken as they stand and the summation evaluated without first windowing the speech segment. This means that E_{min} in Equation (3.8) is minimized over a finite interval, say $0 \leq n \leq N-1$. Hence, Equation (3.10) and Equation (3.11) reduce to

$$\sum_{k=1}^{p} a_k \psi_{ik} = -\psi_{0i}, \qquad 1 \leq i \leq p \tag{3.18}$$

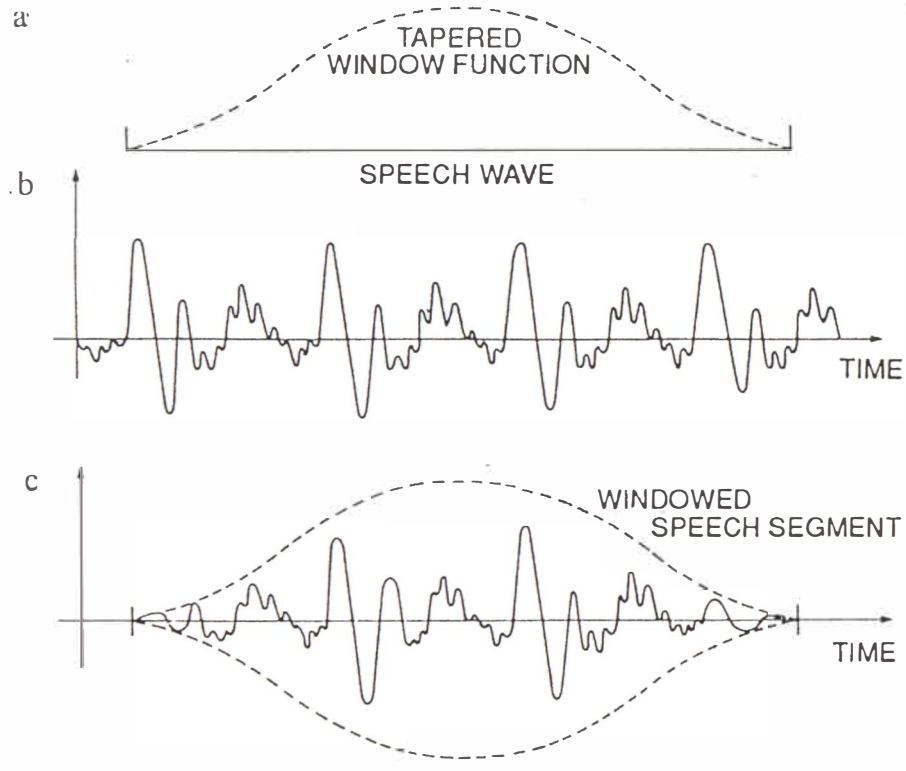


Figure 3.2: *The effects of windowing a segment of speech. (a)The plot of the Hamming window function, (b) The speech waveform to be windowed, (c) The 'windowed' speech. Notice that the ends approach the zero line smoothly.*

Method	Operations	Storage Locations
Gaussian elimination (Kreyszig, 1983)	$p^3/3 + O(p^2)$	p^2
Cholesky decomposition (Kreyszig, 1983)	$p^3/6 + O(p^2)$	$p^2/2$
Durbin-Levinson (Durbin, 1960) (Levinson, 1947)	$p^2 + O(p)$	$2p$

Table 3.1: *Summary of three different algorithms for solving a $p \times p$ matrix.*

$$E_{min} = -\psi_{00} + \sum_{k=1}^{p} a_k \psi_{0k} \tag{3.19}$$

where

$$\psi_{ik} = \sum_{n=0}^{N-1} s_{n-i} s_{n-k} \tag{3.20}$$

is the covariance of the signal s_n in the given interval. The coefficients ψ_{ik} in Equation (3.18) form a $p \times p$ *covariance* matrix. From Equation (3.20), it can be seen that the covariance matrix ψ_{ik} is symmetric, *i.e.* $\psi_{ik} = \psi_{ki}$. However, unlike the autocorrelation matrix, the terms along each diagonal are not equal. Again, a solution can be found by matrix inversion, although in this case the inversion is not simplified by the absence of the Toeplitz condition. However, the covariance method has the advantage that there is no need to multiply the speech segment by a window function.

3.1.3.4 Methods for solving the predictor coefficients

We have seen how the autocorrelation and covariance methods have led to Equation (3.13) and Equation (3.18). From these two (matrix) equations, the predictor coefficients $a_k, 1 \leq k \leq p$ can be computed using several standard methods summarised in Table 3.1. The Gaussian elimination method (Kreyszig, 1983) requires $p^3/3 + O(p^2)$ operations (multiplications or divisions) and p^2 storage locations (see Table 3.1). However, since the autocorrelation matrix $R(i-k)$ in Equation (3.13) and the covariance matrix in Equation (3.18) are symmetric, they can be solved more efficiently using the Cholesky decomposition method (Kreyszig, 1983). As summarised in Table 3.1, the Cholesky method requires only about half the computation ($p^3/6 + O(p^2)$) and about half the storage ($p^2/2$) compared to the Gaussian elimi-

Step 1 Initialization

$$E_0 = R(0)$$

Step 2 Recursion

For $1 \leq i \leq p, \quad 1 \leq j \leq i-1$

$$k_i = -\left[R(i) + \sum_{j=1}^{i-1} a_j^{i-1} R(i-j)\right] / E_{i-1}$$

$$a_i^{(i)} = k_i$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}$$

$$E_i = (1 - k_i^2) E_{i-1}$$

Step 3 Termination

For $1 \leq j \leq p$

$$a_j = a_j^{(p)}$$

Figure 3.3: *Steps in the Durbin-Levinson Algorithm.*

nation method. Further reduction in computational load and storage space can be achieved in the case of the autocorrelation matrix by exploiting its symmetry and the fact that the elements along any diagonal are identical (see Equation (3.15)). This method, often referred to as the Durbin-Levinson algorithm (Durbin 1960; Levinson 1947), requires only $2p$ storage locations and $p^2 + O(p)$ operations. This represent a huge saving compared to the more general Gaussian elimination method. The recursive steps involved in the Durbin-Levinson algorithm is illustrated in Figure 3.3.

It should be emphasized that, in most cases, solving the two matrix equations indicated by Equation (3.13) and Equation (3.18) do not constitute the major computational load. The computation of the autocorrelation or covariance coefficients require pN operations. This can dominate the computation time if $N \gg p$. In most speech (or any other signal) processing applications, this condition is often satisfied.

3.1.3.5 Discussion on the Durbin-Levinson algorithm

The Durbin-Levinson algorithm (see Figure 3.3) is an elegant method for solving the predictive coefficients a_k from the matrix equations Equation (3.13) or its expanded form, Equation (3.15). Referring to Equation (3.15), it is easy to see that if all the autocorrelation coefficients R_i are scaled by a constant, the solution to Equation (3.15) is unchanged. In particular, if the scaling constant is $R(0)$, then the new coefficients, known as the *normalized autocorrelation coefficients* and denoted by $r(i)$ is given by:

$$r(i) = \frac{R(i)}{R(0)} \tag{3.21}$$

Notice that in Equation (3.21), $|r(i)| \leq 1$ because $R(0) \geq R(i)$, $i \neq 0$. The scaling is important especially when the computation is performed in fixed-point arithmetic. This is one of the reason that makes the autocorrelation method the preferred method when compared to the covariance method (Witten, 1982).

Another important feature of the Durbin-Levinson algorithm is that in obtaining the solution for a prediction of order p, one also obtains as a bonus, the solutions, *i.e.* the predictive coefficients a_i, $1 \leq i \leq p$, for all predictors of order less than p and the corresponding minimum predictive error E_i at every step (see Figure 3.3).

The behaviour of E_i is interesting and is worth discussing. From the recursive step in the Durbin-Levinson algorithm, (see Figure 3.3), we know that

$$E_i = (1 - k_i^2)E_{i-1} \tag{3.22}$$

where E_i and E_{i-1}, which have already been defined in Equation (3.11) are the minimum squared error for the predictor of order i and $i-1$ respectively. Rearranging Equation (3.22), we obtain

$$\frac{E_i}{E_{i-1}} = 1 - k_i^2 \tag{3.23}$$

Now, because E_i and E_{i-1} are always positive, *i.e.*

$$0 \leq E_i, E_{i-1} \tag{3.24}$$

This means that the left hand side (LHS) of Equation (3.23) ($\frac{E_i}{E_{i-1}}$) is also positive.

In order for the right hand side (RHS) of Equation (3.23) to satisfy the positivity condition:

$$0 \leq 1 - k_i^2 \tag{3.25}$$

which can only be satisfied if and only if:

$$0 \leq k_i^2 \leq 1 \tag{3.26}$$

The result in Equation (3.26) allows us to supplement Equation (3.25) with an upper bound. Thus,

$$0 \leq 1 - k_i^2 \leq 1 \tag{3.27}$$

which can be substituted back into Equation (3.23) to yield the result:

$$0 \leq E_i \leq E_{i-1} \tag{3.28}$$

From Equation (3.28), we can conclude that the minimum squared error E_i decreases monotonically as the order of the predictor increases. This result has been verified experimentally by Atal and Hanauer (1971). Their result is depicted in Figure 3.4.

3.1.3.6 Comments on filter stability

After the predictive coefficients are computed, the stability of the resulting (all-pole) filter $H(z)$, defined in Equation (3.2), need to be examined. The stability of the filter $H(z)$ is guaranteed if all its poles lie inside the unit circle in the z-plane

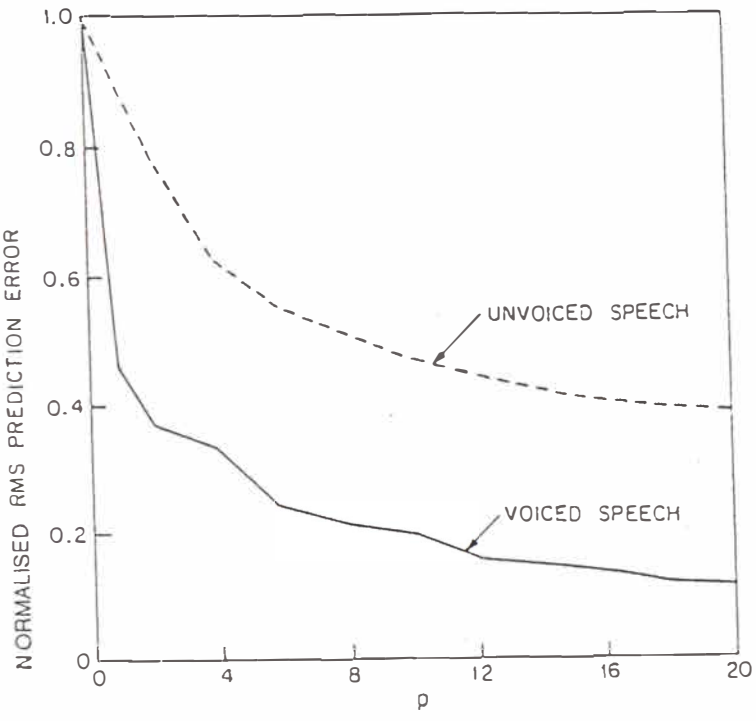


Figure 3.4: *Normalised r.m.s prediction error, plotted against p, the prediction order, for voiced and unvoiced speech. From Atal and Hanauer (1971).*

(Makhoul, 1975). The poles of $H(z)$ are the roots of the denominator polynomial $A(z)$, where

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{3.29}$$

and

$$H(z) = \frac{G}{A(z)} \tag{3.30}$$

$A(z)$ is known as the *inverse filter.*

It has been shown (Grenander and Szegö, 1958) that if the autocorrelation function $R(i)$ is computed from a non-zero signal using Equation (3.17), then the autocorrelation matrix in Equation (3.13) will be positive definite (Kreyszig, 1983). This in turn ensures that the inverse filter coefficients a_k calculated from Equation (3.13) will give rise to a stable $H(z)$.

The positive definiteness of $R(i)$ can often be lost if it is represented by a small word length in a computer. Further, roundoff errors can cause the autocorrelation matrix to become ill-conditioned (Makhoul, 1975). These two conditions can cause instability in $H(z)$. Hence, it is often necessary to check for the stability of $H(z)$. Solving for the roots of $A(z)$ to see if all the roots are inside the unit circle is costly and is best avoided.

Makhoul (1975) stated that a necessary and sufficient condition for the stability of $H(z)$ is for $E_i > 0$, $1 \leq i \leq p$. Therefore, a better alternative for stability check

is to see if all the successive errors are positive. Furthermore, from the derivation given earlier (see Equation (3.22) – Equation (3.28)), it is clear that an equivalent condition for the stability of $H(z)$ is

$$|k_i| < 1, \qquad 1 \leq i \leq p. \tag{3.31}$$

Recall from the Durbin-Levinson algorithm (see Figure 3.3) that k_i, known as the *reflection coefficient*, is one of the intermediate quantities computed in the recursive steps. Thus, the Durbin-Levinson algorithm also facilitates the check for the stability of the filter $H(z)$. The reflection coefficients can also be calculated from the prediction coefficients using a backward recursion procedure (Gray and Markel, 1979):

$$\begin{aligned} k_i &= a_i^{(i)} \\ a_j^{(i-1)} &= \frac{a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)}}{1 - k_i^2}, \quad 1 \leq j \leq i-1 \end{aligned} \tag{3.32}$$

where the index i takes the values $p, p-1, ..., 1$ in that order with $a_j^{(p)} = a_j, 1 \leq j \leq p$.

We have seen that, provided that the positive definiteness of the matrix is not destroyed by the limited word length of a computer, the predictive coefficients resulting from a solution to the autocorrelation matrix equation is guaranteed to form a stable filter. However, the same cannot be said of the predictive coefficients resulting from a solution to the covariance matrix Equation (3.18).

There are several ways of avoiding instability of the covariance method. One way to improve the stability of the covariance method is to use larger number of signal samples, N. This is intuitively correct because as N increases, the covariance matrix approaches an autocorrelation matrix. Another way is to add a very small number to the diagonal elements of the covariance matrix (Makhoul, 1975). However, despite all these precautions, an unstable filter may still result. Fortunately, an unstable (predictive) filter can be made stable by reflecting the poles outside the unit circle inside, in such a way that the magnitude of $H(z)$ remains the same (Atal and Hanauer, 1971).

3.1.4 Computation of the gain

It is now necessary to summarise what has been achieved so far before going any further. Recall that in the all-pole linear prediction model of speech signal, we model the present speech sample, s_n, as a linear combination of the past p samples, s_{n-k}, $k = 1, ..., p$, and the present input u_n (see Equation (3.4)). For reasons that have already been outlined in §3.1.3, the present input is then set to zero, *i.e.* $u_n = 0$ (see Equation (3.4)). This leads to Equation (3.6). Starting from Equation (3.6), it is possible, as we have seen, to derive an optimal set of predictive coefficients, which would minimise the squared error between the predicted present sample $\hat{s}_n$ and the present sample s_n.

Now, because our derivation is based upon the assumption that $u_n = 0$, it may not seem to make sense, at first glance, to determine a value for the gain G (see Equation (3.4)). However, there are interesting observations that can be made.

By rewriting Equation (3.7) as

$$s_n = -\sum_{k=1} p a_k s_{n-k} + e_n \tag{3.33}$$

and then comparing Equation (3.4) and Equation (3.33), we see that for the output to be the signal s_n,

$$Gu_n = e_n \tag{3.34}$$

i.e. the input signal, u_n, must be proportional to the error signal, e_n. For any other input, the output will be different from s_n, which is undesirable. However, if we constrain the energy in the output signal to be the same as the original signal s_n, we can then specify the total energy in the input signal (amplified by the gain G). Since the filter $H(z)$ is fixed, this means that the total energy in the input signal Gu_n must equal the total energy in the error signal, which is given by E_{min} in Equation (3.14) in the case of the autocorrelation method or Equation (3.19) in the case of the covariance method.

The procedure for calculating the input gain G in Equation (3.4) is derived here. The response of the all-pole predictive filter $H(Z)$ to an impulse input is first examined. By applying the energy constraint discussed in the last paragraph, the input gain G is then determined.

Let the input to the all-pole filter $H(z)$ be an impulse at $n = 0$, *i.e.* $u_n = \delta_n$, where δ_n is the Kronecker delta function defined as

$$x(n) = \delta(n) = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.35}$$

The output h_n is then given by

$$h_n = -\sum_{k=1}^{p} a_k h_{n-k} + G\delta_n \tag{3.36}$$

where h_n is the impulse response of $H(z)$.

The autocorrelation $\hat{R}(i)$ of the impulse response h_n is then computed. From the definition (see Equation (3.12)) of the autocorrelation function, $\hat{R}(i)$ is obtained by multiplying Equation (3.36) by h_{n-i} and then summing the product over all n. The result can be shown to be (Makhoul, 1973):

$$\hat{R}(i) = -\sum_{k=1}^{p} a_k \hat{R}(i-k), \qquad 1 \le |i| \le \infty \tag{3.37}$$

$$\hat{R}(0) = -\sum_{k=1}^{p} a_k \hat{R}(k) + G^2 \tag{3.38}$$

Given the condition that the total energy in h_n must equal that in s_n, this means

$$\hat{R}(0) = R(0) \tag{3.39}$$

since the zeroth autocorrelation coefficient is equal to the total energy in the signal. Further, we see that Equation (3.13) and Equation (3.37) are similar. This similarity, together with Equation (3.39) implies that:

$$\hat{R}(i) = R(i), \qquad 0 \le i \le p. \tag{3.40}$$

Finally, from Equation (3.14), Equation (3.38) and Equation (3.40), the gain is given by

$$G^2 = E_{min} = R(0) + \sum_{k=1}^{p} a_k R(k) \tag{3.41}$$

where G^2 is the total energy in the amplified input $G\delta_n$.

3.2 Vector quantization

Quantization is a process whereby signals whose amplitudes are continuous (analog) are approximated by discrete (digital) signals. It is an important aspect of *data compression* or *coding*, the field concerned with the reduction of the number of bits necessary to transmit or store analog data, subject to a *distortion* or *fidelity criterion.*

Depending on the dimensionality of the signal to be quantized, two distinct classes of quantization process can be distinguished. If the signal to be quantized is *one-dimensional*, for example, the temperature reading at a particular point in a room is to be quantized *individually*, then the quantization process is called *scalar quantization.* We have seen the application of scalar quantization process (see §2.2.8) where the signals to be converted is one-dimensional. One the other hand, if a *block* or several of the temperature readings are to be quantized, the process is termed *vector quantization.* Thus, vector quantization is a generalization of scalar process. It is essentially a scalar quantization process in multi-dimensions.

The following discussions begin by introducing the basic concepts of vector quantization (see §3.2.1). An iterative vector quantization algorithm is given in §3.2.2. Various examples of applications of vector quantization are also explored (see §3.2.3. The emphasis of the expositions is on basic principles rather than the elaboration of various techniques and their variations.

The theoretical foundations and performance bounds of various vector quantization techniques have not been included in the following discussions. This is because a discussion on the subject would be highly mathematical and hence would obscure the understanding of its basic principles. Interested readers are referred to the classical paper by Shannon (1948), where the theoretical foundations of vector quantization are laid down. Subsequent theoretical developments of vector quantization have been dealt with adequately in a review article by Makhoul *et al.* (1985), the March, 1982 issue of the IEEE Transactions on Information Theory, which is a special issue on quantization, and a recent book by Jayant and Noll (1984).

3.2.1 Basic concepts of vector quantization

The basic concepts of vector quantization are introduced here. In particular, the concepts of *codebook*, the *level* or *size* of the codebook and *distortion measures* are explained.

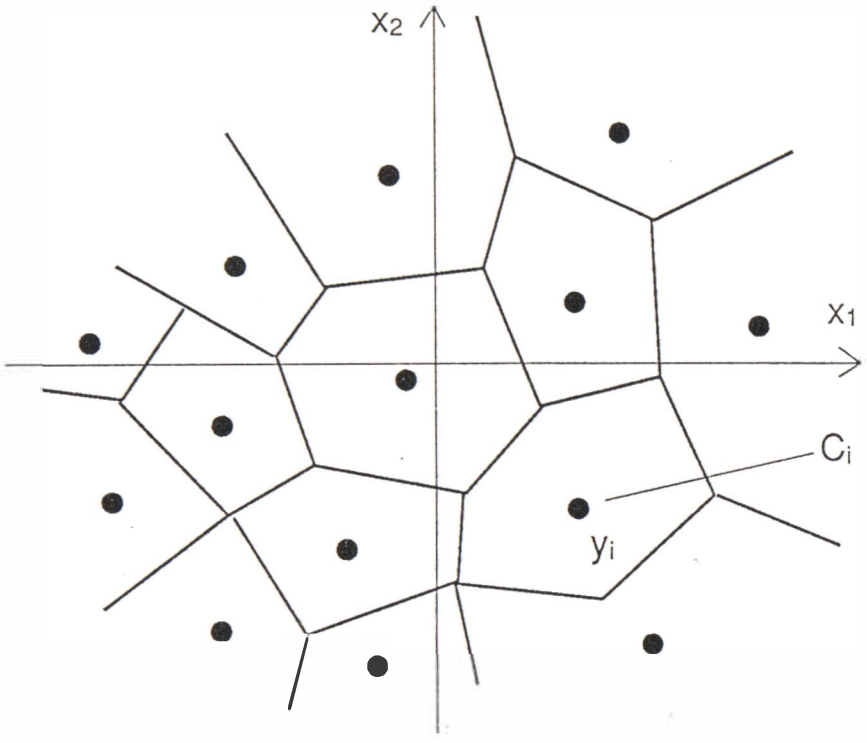


Figure 3.5: *Concepts of the vector quantization process. The two-dimensional space is divided into 14 cells. All input vectors in cell C_i is quantized as the output vector y_i. The cells can be of varying shapes and sizes. After Makhoul* et al. *(1985).*

3.2.1.1 Vector quantization (VQ) codebook and its size

In order to illustrate the concepts of the vector quantization process, a two dimensional (*i.e.* $N = 2$) vector quantizer is considered. Figure 3.5 shows a two-dimensional space which has been divided into 14 cells of various shapes and sizes. Each cell is identified by a label C_i, $1 \leq i \leq 14$. Associated with each cell is a *code vector* y_i, $1 \leq i \leq 14$. The input vectors in this case are two-dimensional vectors $x = [x_1, x_2]$. In vector quantization, each input vector x is mapped into an output vector y. We say that x is quantized as y and y is the quantized value of x. This is written as Equation (3.42)

$$\mathrm{y} = q(\mathrm{x}) \tag{3.42}$$

where $q()$ is the quantization operator.

The set of output vector $Y = \{y_i, 1 \leq i \leq L\}$ is referred to as the codebook, L is known as the codebook size and $\{y_i\}$ are the set of code vectors. Thus, in the example as shown in Figure 3.5, $L = 14$. The codebook size is also frequently referred to as the number of levels.

3.2.1.2 Distortion measures

When a vector x is quantized as another vector y, a quantization error results and a *distortion measure* $d(x,y)$ can be defined between x and y. The distortion mea-

sure $d(x,y)$ is sometimes known as dissimilarity or distance measure (Makhoul *et al.*, 1985). To be useful, a distortion measure must be tractable, so that one can analyzed it and compute it, and be subjectively relevant, so that similarity in distortion measures can be used as an indication of similar speech quality. However, many researchers have found that a few decibels of decrease in a distortion measure is quite perceivable by the ear in one situation but not in another. The careful researcher has learned that, while objective distortion measures are necessary and useful tools in the design of speech coding systems, periodic subjective quality testing is indispensable to making an informed decision on directions for improving system performance. Some of the most commonly used distortion measures are discussed in the following paragraphs.

1) *Mean-Square Error:* The most commonly used distortion measure is the mean-square error (mse) defined as

$$d_2(\mathbf{x},\mathbf{y}) = \frac{1}{N}(\mathbf{x},\mathbf{y})^T(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{k=1}^{N}(x_k - y_k)^2 \tag{3.43}$$

where the distortion is defined per dimension. The main attraction of the mse as a distortion measure is its simplicity and mathematical tractability. A more general distortion measure based on the L_r norm is defined as

$$d_r(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{k=1}^{N}|x_k - y_k|^r \tag{3.44}$$

Note that Equation (3.44) is equal to Equation (3.43) for $r = 2$. The other two most popular values of r are $r = 1$ and $r = \infty$.

For the mse distortion, it is common practice to measure the performance of a coding scheme by the signal-to-noise ratio (or signal-to-quantization-noise ratio)

$$SNR = 10\log_{10}\frac{E(||\mathbf{x}||^2)}{E\left[d(\mathbf{x},\mathbf{y})\right]} \tag{3.45}$$

which corresponds to the ratio of the average energy to the average distortion or quantization noise. Notice that the SNR is in logarithmic scale and the unit is in dB.

2) *Weighted Mean-Square Error:* The mse distortion criterion can be further generalised by introducing input-dependent weightings. While not subjectively meaningful in many cases, these input-dependent weightings have proved useful and only slightly more complicated (Gray, 1984). A general weighted mse is defined as

$$d_w(\mathbf{x},\mathbf{y}) = (\mathbf{x},\mathbf{y})^T\mathbf{W}(\mathbf{x},\mathbf{y}) \tag{3.46}$$

where $\mathbf{W}$ is a positive-definite weighting matrix. One popular choice for the weighting matrix is $\mathbf{W} = \mathbf{\Gamma}^{-1}$, where $\mathbf{\Gamma}$ is the covariance matrix of the input vector $\mathbf{x}$

$$\Gamma = E\left[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\right] \tag{3.47}$$

and

$$\bar{\mathbf{x}} = E\left[\mathbf{x}\right] \tag{3.48}$$

As it turns out, by defining the weighting matrix $\mathbf{W}$ as the inverse of the covariance matrix, $\mathbf{\Gamma}^{-1}$, (see Equation (3.46) - Equation (3.49)), the matrix $\mathbf{W}$ is also symmetric as well as being positive definite. This means that $\mathbf{W}$ can be factorized as

$$\mathbf{W} = \mathbf{P}^{\mathbf{T}}\mathbf{P} \tag{3.49}$$

By transforming the vectors $\mathbf{x}$ and $\mathbf{y}$ into a new set of vectors $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ where

$$\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x} \qquad \tilde{\mathbf{y}} = \mathbf{P}\mathbf{y} \tag{3.50}$$

and a few mathematical manipulations, Makhoul *et al.* (1985) have shown that,

$$d_w(\mathbf{x}, \mathbf{y}) = d_2(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \tag{3.51}$$

Thus, the weighted mse between the original vectors is equal to the mse between the transformed vectors. Therefore, for computational purposes, it may be advantageous to perform the transformation in Equation (3.50) on all the data before vector quantization is performed.

3) *Saito-Itakura Distortion Measure:* Another distortion measure was proposed by Itakura (1975). Known as the Itakura-Saito distortion measure, it was based on the minimum-likelihood principles. The Itakura-Saito distortion measure is defined as

$$d_I(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{R}_{\mathbf{x}} (\mathbf{x} - \mathbf{y}) \tag{3.52}$$

where

$$\mathbf{R}_{\mathbf{x}} = \{R(i-k)/R(0), \;\; 0 \leq i, k \leq N-1\} \tag{3.53}$$

is the normalised autocorrelation matrix whose coefficients $R(i-k)$ were used in computing the vector of predictor coefficients $\mathbf{x} = [a_1 a_2 \cdots a_N]^T$ in Equation (3.13). It is interesting to note that $\mathbf{R}_{\mathbf{x}}$ is effectively a weighting matrix. However, there are some subtle differences between $\mathbf{R}_{\mathbf{x}}$ and the weighting matrix $\mathbf{W}$ in Equation (3.46). For example, $\mathbf{R}_{\mathbf{x}}$ has a one-to-one relationship with $\mathbf{x}$ and thus changes value as $\mathbf{x}$ changes, in contrast with $\mathbf{W}$ which is fixed. Furthermore, the Itakura-Saito distortion measure is not symmetric with respect to its arguments, *i.e.* $d_I(\mathbf{x}, \mathbf{y}) \neq d_I(\mathbf{y}, \mathbf{x})$.

4) *Perceptually Related Distortion Measures:* Because coding at high bit rate (32 kbps and above) introduces very little distortion or quantization, most reasonable distortion measures, including those discussed above, correlate well with subjective judgements of speech quality. However, as the bit rate decreases and the distortion increases, simple distortion measures have not always correlated well with perceptual judgements. A number of perceptually based distortion measures that correlate well with subjective judgements have been used for speech coding. Examples of these distortion measures are the segmental SNR which is the mean value of SNR calculated at every $20 - 30ms$, using real speech as the test signal (Kitawaki *et. al.* 1988; Thorpe 1990).

Another successful perceptually related distortion measure is the LPC Cepstrum Distance (CD) measure (Kitawaki *et al.*, 1984). It is effectively the difference between the spectrum envelopes of the input and the output (which is the coded) speech

signals. The CD values are calculated by cepstrum analysis as follows

$$CD = 10 \log \sqrt{2 \sum_{k=1}^{p} \{C_x(k) - C_y(k)\}^2} \tag{3.54}$$

where $C_x(k)$ and $C_y(k)$ are the LPC cepstrum coefficients of the input and output coding system signal, and p is the order of the coefficients.

Kitawaki *et al.* (1984) conducted some extensive studies to compare several perceptually related distortion measures. From these studies, they concluded that

1. Objective distortion measures in the frequency domain, such as CD, have better correspondence to the subjective judgemental values than that in the time domain, such as the segmental SNR.
2. Among the several objective distortion measures in the frequency domain, CD measure has the best correspondence to subjective judgemental values.

3.2.2 Codebook design

As mentioned before, in order to quantize a set of N-dimensional vectors into L discrete vectors, one needs to partition the N-dimensional space into L cells and associate each cell C_i, $1 \leq i \leq L$ with a code vector $\mathbf{y_i}$. An input vector $\mathbf{x}$ is then assign the code vector $\mathbf{y_i}$ if it lies inside the cell C_i.

The purpose of a codebook design algorithm is to derive, from a set of N-dimensional training vectors $\{\mathbf{x}(n), 1 \leq n \leq M\}$, another set of N-dimensional code vectors $\mathbf{y_i}$, which is optimal in the sense that a pre-defined distortion measure $d(\mathbf{x}, \mathbf{y})$ is minimized.

The quantizer is said to be optimal (minimum-distortion) if two conditions are satisfied. The first condition is that the quantizer must use a minimum distortion or nearest neighbour selection rule

$$q(\mathbf{x}) = \mathbf{y_i} \ \ \textit{iff} \ \ d(\mathbf{x}, \mathbf{y_i}) \leq d(\mathbf{x}, \mathbf{y_j}), \ \ j \neq i, 1 \leq j \leq L. \tag{3.55}$$

That is, the quantizer assigns the code vector that results in the minimum distortion with respect to $\mathbf{x}$. The second necessary condition for optimality is that each code vector is chosen so that the average distortion in cell C_i is minimized. In other words, $\mathbf{y_i}$ is chosen so that

$$D_i = E\left[d(\mathbf{x}, \mathbf{y_i}) | \mathbf{x} \in C_i\right] = \int_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{y_i}) p(\mathbf{x}) d\mathbf{x} \tag{3.56}$$

is minimized. The code vector $\mathbf{y_i}$ is also known as the *centroid* of cell C_i while the cells thus defined are known as the nearest neighbour cells, Voronoi cells, or Dirichlet regions (Gersho, 1982).

A popular codebook design algorithm which satisfies the two optimality criteria discussed above is the generalized Lloyd algorithm (Lloyd, 1982). The algorithm *iteratively* improves on an initial set of code vectors. Two basic variations of the

Step 1 Initialization:

Set the iteration index $m = 0$. Choose a set of initial code vectors $\mathbf{y_i}(m)$, $1 \leq i \leq L$

Step 2 Classification:

Classify the set of training vectors $\{\mathbf{x}(n), 1 \leq n \leq M\}$ into the cluster C_i by the nearest-neighbour rule.

Step 3 Codebook Updating: $m \leftarrow m + 1$.

Update the codebook of every cluster by computing the centroid of the training vectors in each cluster.

Step 4 Termination Test:

If the decrease in the overall distortion $D(m)$ at iteration m relative to $D(m-1)$ is below a certain threshold, stop; otherwise go to step 2.

Figure 3.6: *Steps in the K-Means Vector Quantisation Algorithm.*

generalized Lloyd algorithm, each employing a different method of choosing an initial set of code vectors, have been developed. These two methods are known as the K-Means algorithm (Rabiner *et al.*, 1986) and the LBG algorithm (Linde *et al.*, 1980) and are illustrated in Figure 3.6 and Figure 3.7 respectively.

3.2.3 Applications of vector quantization (VQ)

As described earlier, the function of a vector quantization system is to map a sequence of continuous or discrete vectors into a digital sequence suitable for communication over or storage in a digital channel. Hence, The main goal of such a system is data compression: to reduce the bit rate so as to minimize communication channel capacity or digital storage requirements while maintaining the necessary fidelity of the data.

The two forms of data whereby data compression techniques such as vector quantization are most frequently employed are speech signals and video images. In the coding of speech signals, for example, systems employing VQ technique and are capable of reproducing good quality speech at 800 bps have been developed (Wong *et al.*, 1982). This represents a significant reduction in the bit rate previously re-

Step 1 Initialization:

Set the iteration index $m = 0$. Compute the centroid of the set of training vectors $\{\mathbf{x}(n), 1 \leq n \leq M\}$

Step 2 Splitting:

Split each centroid $\mathbf{y_i}(m)$ into two centroids. This doubles the total number of centroids.

Step 3 Classification:

Classify the set of training vectors into the clusters by the nearest-neighbour rule.

Step 3 Codebook Updating: $m \leftarrow m + 1$

Update the codebook by computing the centroid of the training vectors in each cluster.

Step 4 Termination Test:

If the decrease in the overall distortion $D(m)$ at iteration m relative to $D(m-1)$ is below a certain threshold *and* the number of clusters has reached the required *level* L , stop; otherwise go to step 2.

Figure 3.7: *Steps in the LBG Vector Quantisation Algorithm.*

quired for comparable quality. In addition, speech coding systems which perform in real-time have also been implemented in specialized digital signal processing chips (Bukiet *et al.*, 1987). In the coding of video images, successful applications of VQ techniques have been demonstrated by Gersho and Ramamurthi (1982) and Murakami *et al.* (1982).

Vector quantization techniques have also found successful applications in isolated word recognition system (Rabiner *et al.*, 1983), speaker recognition system (Soong *et al.*, 1987).

3.3 Summary

The essential mathematical background of two very important speech processing techniques have been outlined. The first technique is known as the LPC technique. It assumes that the speech to be processed can be adequately modelled by the generalized linear prediction model. By introducing some simplifications to the generalized linear prediction model, the all-pole linear prediction model results. The all-pole linear prediction model is completely characterised by a set of linear predictive coefficients and a gain or energy term. Based on the all-pole linear prediction model, a set of matrix equations is then derived. The Durbin-Levinson algorithm for solving this set of equations is then discussed. Comments on the resulting all-pole linear prediction model have also been offered.

Vector quantization is the second technique that has been discussed. The concepts of codebook, codebook size or VQ level, distortion measure have been treated. Two variations of the generalized Lloyd algorithm for designing a codebook are then outlined. The discussion on vector quantization concludes with a survey of the applications of vector quantization techniques.

Chapter 4

TECHNIQUES FOR AUTOMATIC SPEECH RECOGNITION

"The cosmic religious experience is the strongest and noblest driving force behind scientific research."
(Albert Einstein. Quoted in an obituary on 19 Apr 1955)

4.1 Introduction

Automatic speech recognition is fundamentally a pattern classification task. The object is to take an unknown input pattern of a speech waveform, and classify it as one of a set of pre-determined vocabulary. The recognition task typically involves a two-step process (see Figure 4.1). The first step involves extracting, from the input speech waveform, a time sequence of features that characterise the input. The second step is pattern classification where the sequence of features is compared against the speech recogniser's stored knowledge of its vocabulary, and a decision rule is applied to arrive at a transcription of the input utterance.

4.2 Feature extraction

The most important function of the feature extraction step is to reduce the amount of storage space of the data to be processed to a manageable size. This is achieved by removing the background noise, channel distortion, speaker characteristics (in the case of speaker-independent speech recognition system), and manner of speaking

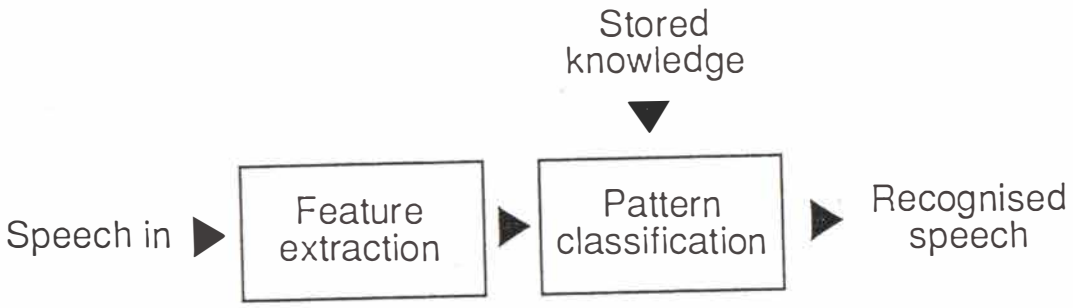


Figure 4.1: *Speech recognition as a two-step pattern classification task.*

from the raw speech waveform and retaining only those parts that are necessary for recognition. The speech waveform is usually divided into overlapping time frames of 10-20 ms long and a feature vector F is computed for each time frame. The features may be as simple as the energy (or rms) and zero-crossing rate of the waveform during each time frame. Other features that are frequently used are based on the source-filter model of speech production (see Chapter 1). Examples of these features are the LPCs (Lim *et al.*, 1990), cepstral coefficients (Clark *et al.*, 1990) and formant frequencies which have already been discussed in Chapter 2. Features that are derived by modelling the human auditory system or perception have also been used (Hermansky, 1987). This type of analysis usually begins with a set of overlapping band-pass filters, then includes other non-linear effects that occur in auditory processing.

Vector quantization (see Chapter 3) is also commonly used to reduce the amount of storage space and subsequent processing. Strictly speaking, it is not a feature extraction technique. Although it can be used directly on the raw speech waveform, the vector quantization process is usually applied to multi-dimensional features (such as LPCs). Many examples of the use of this technique can be found in the numerous references that have been cited in this thesis. Though vector quantization is useful, it unavoidably introduces some distortion into the speech to be recognised. However, with vector quantization levels of 256 and above, the distortion is negligible.

4.3 Speech pattern classification techniques

The discussion in §2.3 on the historical development of speech recognition research has shown that a myriad of algorithms have been devised to tackle the speech recognition or speech pattern classification task. These algorithms can be conveniently grouped into three simple pattern classification techniques. The Dynamic Time Warping (DTW) technique (Furui 1981; Clark *et al.* 1990; Sakoe and Chiba 1978) is presented in §4.3.1 while the Hidden Markov Modelling (HMM) technique (Rabiner and Juang 1986; Poritz 1988; Rabiner 1989) and the Neural Network technique are discussed in §4.3.2 and §4.3.3 respectively.

4.3.1 Dynamic time warping technique

4.3.1.1 Time alignment problem

Let us denote the time sequence of the feature vectors extracted from a test word and a reference word as

$$T(n) = t_1, t_2, t_3, \cdots, t_N \tag{4.1}$$

and

$$R(m) = r_1, r_2, r_3, \cdots, r_M \tag{4.2}$$

respectively. $T(n)$ and $R(m)$ are referred to as the test pattern and reference template from now on.

In general, the two sequences $T(n)$ and $R(m)$ will not have the same length, *i.e.* $N \neq M$, because of the variations in speaking speed. This poses a problem for a pattern classifier which is required to compute the distance between the test pattern and the reference template, as it will not normally be possible to align the endpoints of the two patterns.

4.3.1.2 Solutions

The simplest answer to the time-alignment problem is to linearly expand or contract the time axis of the test pattern to match the length of the reference template. This linear time normalisation technique has been found to improve the recognition score of isolated digits (Denes and Mathews, 1960). One disadvantage of linear time normalisation is that, although it allows the endpoints to be aligned, it does not guarantee that the best match between the internal features of the words will be achieved (Ainsworth, 1988).

A more general solution to the time-alignment problem is to use a non-linear time normalisation technique as shown in Figure 4.2. Figure 4.2a shows a reference template and a test pattern. While both patterns have a lot in common (such as the shapes and relative heights of the peaks), the patterns are different in length and the peaks are not aligned. By means of an optimum time-alignment path (see Figure 4.2b), the peaks may be aligned and a meaningful distance measure can be computed between the two patterns along this optimum path. Hence, as illustrated, non-linear time normalisation effectively 'warps' one pattern to achieve maximum

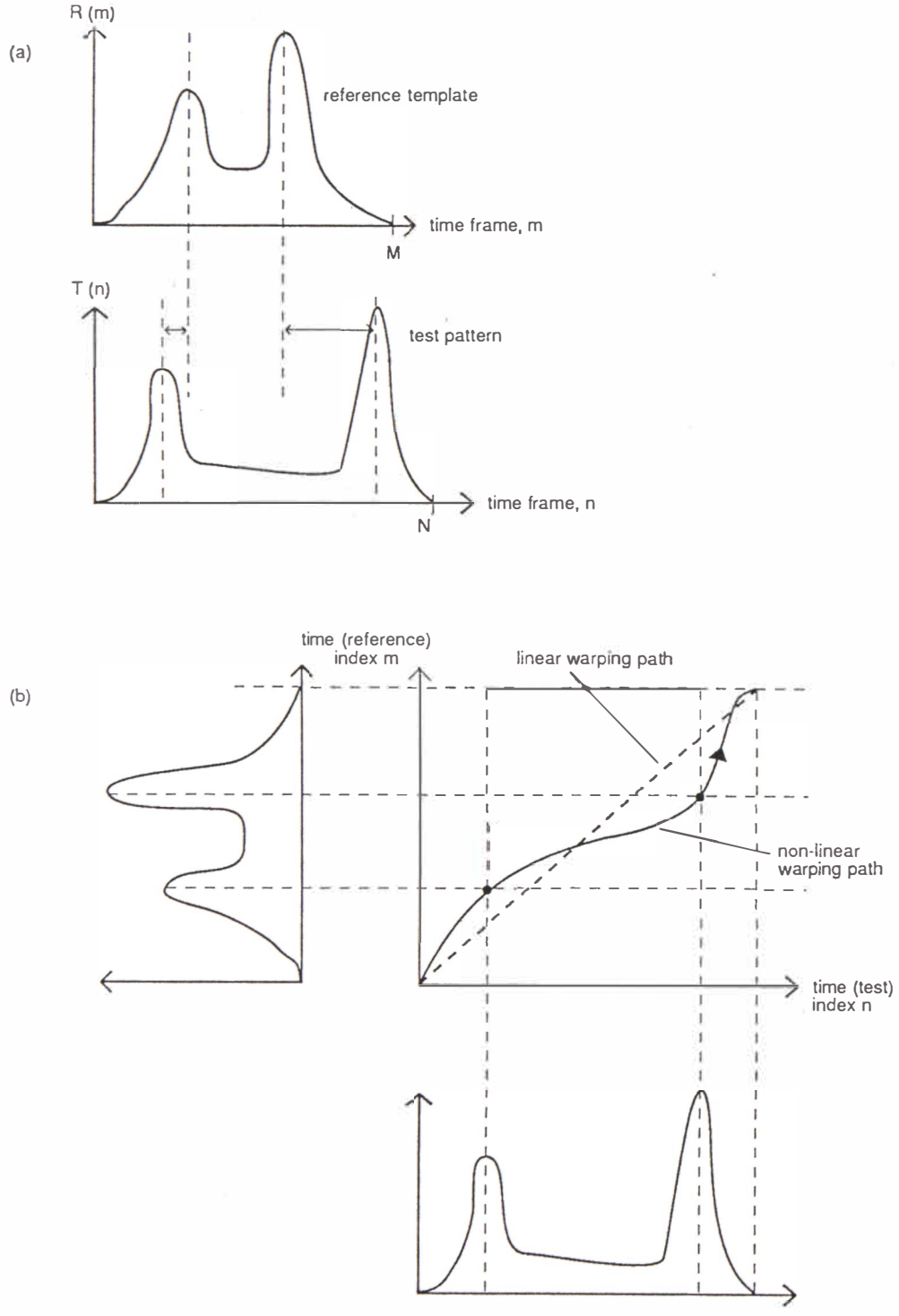


Figure 4.2: *Aligning two patterns of feature vectors non-linearly. (a) A reference template and a test pattern showing the effects of variation in speaking rate. (b) By means of a non-linear warping path, the two patterns are time-aligned.*

correlation with (or least distance from) the other pattern. For this reason, the time-alignment path is also known as the warping path or the warping function.

The warping function is effectively a mapping between the time indices n and m of the two patterns $T(n)$ and $R(m)$ such that time normalisation between them is achieved. The warping (or mapping) function w, between n and m is denoted

$$m = w(n) \tag{4.3}$$

The function w must satisfy a set of boundary conditions at the endpoints and some restrictions on the form it assume. Typically, a warping function must be continuous, monotonically increasing and a slope which is neither too steep nor too gentle (Sakoe and Chiba, 1978). The results of many researchers (Sakoe and Chiba 1978; Rabiner *et al.* 1978; Furui 1981; Clark *et al.* 1990) have confirmed that these constraints accurately model the behaviour of the time sequence of feature vectors computed from human speech.

4.3.1.3 A dynamic programming example

As stated in §4.3.1.2, the optimum warping function can be solved using the dynamic programming technique (Clark *et al.*, 1990). The Dynamic Programming technique is described in more detail by working through an example problem with some simple boundary conditions and constraints.

We will use the reference templates $R(m)$ and $T(n)$ again. Suppose we specify that the warping function must satisfy the following conditions:

1. Boundary Conditions:

$$w(1) = 1, \quad w(N) = M \tag{4.4}$$

and

2. Continuity Conditions:

$$w(n+1) - w(n) = \begin{cases} 0, 1, 2, & w(n) \neq w(n-1) \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

Equation (4.4) states that the endpoints of the $R(m)$ and $T(n)$ must coincide while Equation (4.5) means that the warping function must be monotonically increasing, with a maximum slope of 2, and a minimum slope of 0, except when the slope at the preceding frame was 0, in which case the minimum slope is 1. Figure 4.3 shows an example of time warping function

The next step in dynamic programming involves defining, for every pair of points (n, m) within the parallelogram of Figure 4.3, a suitable distance measure D. Given the distance function D, the optimum dynamic path/function w is chosen to minimize the accumulated distance D_T along the path, *i.e.*

$$D_T = \min_{w(n)} \sum_{n=1}^{N} D(R(n), T(w(n))) \tag{4.6}$$

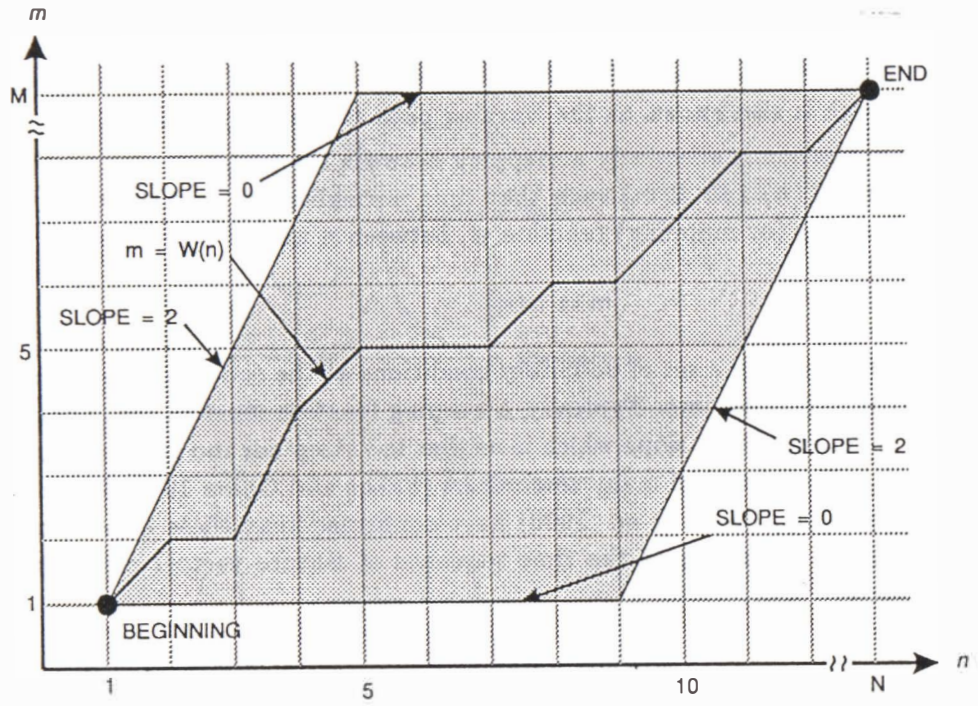


Figure 4.3: *An example of time-warping function which satisfies the conditions specified in Equations (4.4) and (4.5). The parallelogram shows the region of other possible solutions.*

D_T can be determined by solving, in a recursive manner, the accumulated distance to any pair of allowable grid point (n, m) denoted as

$$D_A(n,m) = D(n,m) + \min_{q \leq m} D_A(n-1,q) \tag{4.7}$$

where $D_A(n,m)$ is the minimum accumulated distance to the grid point (n,m) and is of the form

$$D_A(n,m) = \sum_{p=1}^{n} D(R(p), T(w(p))) \tag{4.8}$$

Given the continuity constraint of Equation (4.5), Equation (4.8) can be rewritten as

$$\begin{aligned} D_A(n,m) &= D(n,m) + \\ &\quad \min\left[D_A(n-1,m)g(n-1,m), D_A(n-1,m-1), D_A(n-1,m-2)\right] \end{aligned} \tag{4.9}$$

where $g(n,m)$ is a weighting function of the form

$$g(n,m) = \begin{cases} 1 & w(n) \neq w(n-1) \\ \infty, & \text{otherwise} \end{cases} \tag{4.10}$$

The final solution D_T of Equation (4.6) is, by definition,

$$D_T = D_A(N,M) \tag{4.11}$$

4.3.1.4 Comments

The Dynamic Time Warping (DTW) technique has proved to be an effective technique in removing the effects of variabililty in speaking speed. This has made DTW a popular technique in speech recognition research. Although most of the research published in the literature are on the recognition of English and Japanese words, the DTW technique should be suitable for the recognition of other languages as well. Furthermore, the boundary conditions and the constraints of the warping function can be adjusted to suit a particular application.

Although the ability of DTW technique in ignoring differences (due to variations in speaking speed) in the durations of parts of words has been the main reason for its success, this same ability can sometimes cause confusion between words such as 'league' and 'leek' where the principal distinguishing factor *is* the duration of the vowel. Moore *et al.* (1982) showed that by measuring the local time scale variability and incorporating this knowledge as an additional constraint in the dynamic time warping algorithm a considerable reduction in recognition error can be achieved.

A limitation of the template matching approach is that any template, at any particular time frame, represents only one speech sound. The Hidden Markov Model (HMM) approach creates a more general model of speech using statistical methods.

4.3.2 Hidden Markov modelling (HMM) technique

4.3.2.1 Introduction

The Hidden Markov Model (HMM) (Lim 1990; Picone 1990) is a statistical approach to speech recognition. In this approach, a set of training speech data is used to generate a probabilistic model that best characterizes the training data. The training data may consists of isolated words (Lim *et al.*, 1990), or subwords (Rabiner, 1989).

In the Hidden Markov Model, speech is modelled as a *two stage probabilistic process.* In the first part of the two-stage process, speech is modelled as a sequence of transitions through *states.* These state transitions are governed by a state transition probabilities matrix $A = \{a_{ij}\}$, where a_{ij} is the probability of transiting from state i to state j, given that the present state is state i. Furthermore, the states are *not* directly observable, but manifest themselves by a sequence of observations or features which may assume continuous (Rabiner, 1989) or discrete (Juang, 1985) values. The observations (or features) in any state are not deterministic, but are specified by a probability density function over the space of features. This probability density function is known as the *observation probabilities matrix* denoted by $B = \{b_{jk}\}$, where b_{jk} is the probability of observing symbol k, given that the model is in state j.

The HMM technique was introduced in the late 1960s (Baum and Petrie, 1966) and extended in the early 1970s (Baum *et al.*, 1970). The richness in its mathematical structure gives the technique power and flexibility in modelling many real-life processes (Rabiner, 1989).

The organization of the rest of the discussion on HMM is as follows. In §4.3.2.2, the idea of a discrete *observable* Markov process is outlined. The extension to *Hid-*

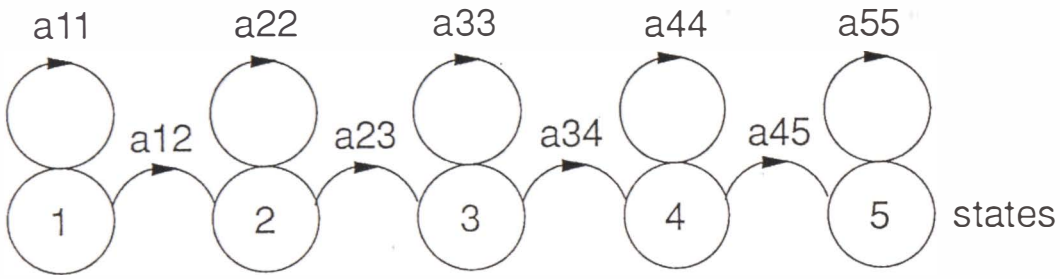


Figure 4.4: *A 5-state Markov process with the associated allowable state transitions.*

den Markov models is provided in §4.3.2.3. The elements of an HMM are then summarised in §4.3.2.4. Given the form of the HMM described in §4.3.2.4, there are three basic problems of interest that must be solved for the model to be useful in real-world applications. These three problems are posed in §4.3.2.5 and their solutions presented in §4.3.2.6. Other types of HMM that have been studied and applied to speech recognition are reviewed in §4.3.2.7.

4.3.2.2 Discrete observable Markov process

Figure 4.4 shows an example of a discrete Markov process. As shown in Figure 4.4, a Markov process may be in any one of N discrete states, labelled $1, 2, \cdots, N$ (where $N = 5$ for simplicity) at a particular instant, t, in time. At regularly spaced discrete times, the system transits to the next state (which may be the same as the present state) according to a set of probabilities associated with each state. For example, the probability a_{12} in Figure 4.4 is the probability that the Markov process will transit to state 2 at time $t + 1$, given that the present state (at time t) is in state 1. The example shown in Figure 4.4 is a *first* order Markov process because the *conditional* probabilities a_{ij}, $1 \leq i, j \leq 5$ is only dependent on *one* previous state , *i.e.* the state at time t.

We will consider only the first order Markov process from now on. We denote the set of state transition probabilities as $A = \{a_{ij}\}$ where

$$a_{ij} = P\left[q_t = j | q_{t-1} = i\right] \tag{4.12}$$

In Equation (4.12), q_t is the state at time t, q_{t-1} is the state at time $t-1$. Further, the probabilities a_{ij} must also obey the standard stochastic constraints (Juang *et al.*, 1985):

$$a_{ij} \geq 0 \tag{4.13}$$

and

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{4.14}$$

The above stochastic process is called an *observable* Markov process since it produces a set of states at each instant of time, where each state corresponds to a physical (observable) event. The following coin toss experiment will demonstrate the ideas developed so far succintly.

Example Of A Discrete Markov Process: Coin Toss Experiment

The *outcome* of a coin toss experiment can be either one of two *events*: (H)ead or (T)ail. I assign these two possible events to one of the following two states:

S_1, State 1 : Head
S_2, State 2 : Tail

I further postulate that the outcome of the toss can be accurately modelled as a discrete Markov process where the state transition probabilities matrix A is given by

$$A = \{a_{ij}\} = \begin{bmatrix} 0.2 & 0.8 \\ 0.1 & 0.9 \end{bmatrix} \tag{4.15}$$

Given that the first toss of the coin is a head, we can ask the question: What is the probability (according to the model) that the outcome for the next three tosses will be "tail-tail-head"?

Since the heads and tails correspond to state 1 and state 2 (see earlier definition) respectively, this sequence of heads and tails can be represented as a sequence of state 1 and state 2. By denoting this observation sequence O as $O = \{S_1, S_2, S_2, S_1\}$, the question can be re-posed as: What is the probability of observing the sequence O, given the model? This probability can be expressed (and evaluated) as

$$\begin{aligned} P(O|\text{model}) &= P(S_1, S_2, S_2, S_1|\text{model}) \\ &= P(S_1) \cdot P(S_1|S_2) \cdot P(S_2|S_2) \cdot P(S_2|S_1) \\ &= \pi_1 \cdot a_{12} \cdot a_{22} \cdot a_{21} \\ &= 1 \cdot (0.8) \cdot (0.9) \cdot (0.1) \\ &= 0.072 \end{aligned} \tag{4.16}$$

where

$$\pi_i = P(O_1 = S_i), \quad 1 \leq i \leq N \tag{4.17}$$

is the initial state probabilities.

Another interesting question that can be asked (and answered) using the model is: Given that the model is in a known state i, what is the probability that it stays in that state for exactly $t = T$ times? The solution is evaluated as follows:

$$\begin{aligned} P(O = \underbrace{S_i, S_i, S_i, \cdots, S_i}_{T}, S_j \neq S_i | \text{model}) &= P(O|\text{model, initial state} = S_i) \\ &= a_{ii}^{(T-1)} \cdot (1 - a_{ii}) \\ &= P_i(t = T) \end{aligned} \tag{4.18}$$

The quantity $P_i(t = T)$ is the (discrete) probability density function of duration T in state i. This exponential duration density is characteristic of the state duration in a Markov chain. Based on $P_i(t = T)$, one can easily calculate the expected number of observations (duration) in a state, conditioned on starting in that state as

$$\begin{aligned} \mathrm{E}(t_i) &= \sum_{t=1}^{\infty} t \cdot P_i(t) \\ &= \sum_{t=1}^{\infty} t \cdot (a_{ii})^{(t-1)} \cdot (1 - a_{ii}) \\ &= \frac{1}{1 - a_{ii}} \end{aligned} \tag{4.19}$$

Hence, for example, the expected number of *consecutive* heads, $\mathrm{E}(t_1)$, according to the model, is $\frac{1}{(1-0.2)} = 1.25$. Similarly, the expected number of consecutive tails, $\mathrm{E}(t_2)$, is $\frac{1}{(1-0.9)} = 10$.

4.3.2.3 Extension to HMM

As outlined in §4.3.2.2, in a discrete Markov process (model), each state *corresponds directly* to an observable (physical) event. The Coin Toss Experiment (see §4.3.2.2) is an example of a discrete Markov process (model). However, this model is too restrictive to be applicable to many problems of interest. In this section, the concept of a discrete Markov model is extended to include the case where *the observation is a probabilistic function of the state*. In other words, the resulting model (which is called a Hidden Markov model) is a doubly embedded stochastic process with an underlying stochastic process that is *not* observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations. To understand further the definition of a HMM, let us consider the following urn and ball model (Rabiner and Juang, 1986).

Urn and Ball Model (Rabiner and Juang, 1986)

Figure 4.5 shows N large urns, each containing finite number of coloured balls. There are M possible different colours. The experiment then proceeds as follows. You are in a room, and according to some random process, you choose an initial urn. From this urn, you pick a coloured ball at random and record the colour of the ball. The ball is then returned to the urn from which it is selected. A new urn is then selected

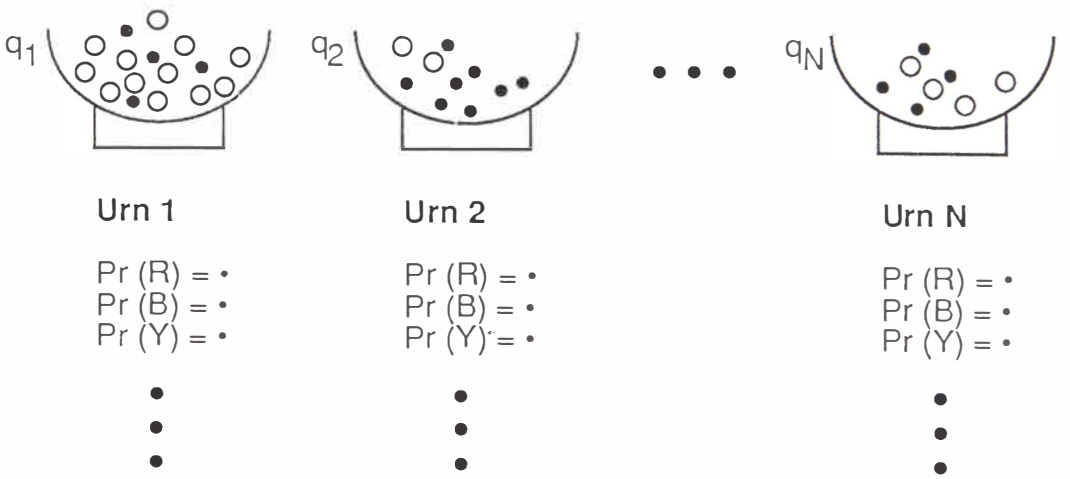


Figure 4.5: *An urn and ball model illustrating the basic concepts of a discrete symbol hidden Markov model. Each urn corresponds to a state in the model. Inside each urn is a large number of coloured balls where each colour represents a symbol. See the text for further details.*

Time Clock	1	2	3	···	T
Urn Sequence (not observed)	3	1	N	···	5
Colour Sequence (observed)	(R)ed	(B)lue	(Y)ellow	···	(G)reen

Table 4.1: *A possible observation sequence of colours from the urns and balls model.*

according to the random selection process associated with the current urn, and the ball selection is then repeated for, say T times. This process generates an observation colour sequence of length T, which is regarded as the observable output of a HMM. A possible outcome of the experiment is illustrated in Table 4.1.

In the urn and ball process that I have just described, each urn corresponds to a state in a HMM. For each urn (state), a colour (observation) probability is defined. In addition, the choice of urns is dictated by the state transition probability matrix of the HMM.

4.3.2.4 Elements of HMM

The observation sequence O of coloured balls (see Table 4.1) resulting from the urns and balls model presented in §4.3.2.3 is an example of how an observation sequence is generated by a HMM. This section will formally define the elements of a HMM, with specific reference to the urns and balls model that has just been outlined in §4.3.2.3. For consistency, I have adopted the same set of notations as that used by Rabiner and Juang (1986) and Rabiner (1989), upon which the following materials (§4.3.2.4 - §4.3.2.6) are largely based.

A HMM is fully specified by the following five parameters:

1. The number of states, N, in the model. Although the states are not observed, some physical significance can often be attached to the states. This is true for many real-life applications. For example, in the urns and balls model, the states correspond to the urns and hence, by extension, N represents the number of urns. By 'uncovering' the hidden states (*i.e.* urns), we can find out from which urn each ball has been extracted. The problem of 'uncovering' the hidden states will be posed in §4.3.2.5 and the solution will be provided in §4.3.2.6. In general, the states are linked together so that each state can be reached from any other state (including itself). However, as shall be seen later, other possible linkage of states are often more interesting. The set of possible states is denoted by $S = \{S_1, S_2, \cdots, S_N\}$ while the state at time t by q_t.
2. The number of observation symbols, M, in each state. Each observation symbol correspond to the physical, observable output of the system being modelled. For the urn and ball model, the *colours* of the balls are the symbols. The set of possible symbols is denoted by $V = \{V_1, V_2, \cdots, V_M\}$.
3. The state transition probability matrix $A = \{a_{ij}\}$, where

$$a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right], \quad 1 \leq i, j, \leq N \tag{4.20}$$

is the probability of transiting to state S_j at time $t+1$, given that the state at time t is S_i. For the general case where any state can reach any other state in a single step, $a_{ij} > 0$ for all possible pairs of (i, j). In all other cases, $a_{ij} = 0$ for at least one (i, j) pair.

4. The observation symbol probability matrix $B = \{b_{jk}\}$, where

$$b_{jk} = P\left[V_k \text{ at } t | q_t = S_j\right], \quad \begin{array}{l} 1 \leq j, \leq N \\ 1 \leq k, \leq M \end{array} \tag{4.21}$$

is the probability of observing symbol V_k, given that the present state q_t is S_j.

5. The initial state probability matrix $\pi = \{\pi_i\}$, where

$$\pi_i = P\left[q_1 = S_i\right], \quad 1 \leq i, \leq N \tag{4.22}$$

is the probability of starting at state S_i, at time $t = 1$.

In summary, specification of a HMM involves choice of the number of states, N, the number of discrete symbols, M, and the specification of the three probability matrices: A, B and π for each word. The compact notation,

$$\Lambda = (A, B, \pi) \tag{4.23}$$

will be used, from now on, to represent a HMM.

4.3.2.5 The three basic problems for a HMM

Before we consider the three basic problems for a HMM, let us suppose that appropriate values have been assigned to the parameters N, M, A, B and π of a HMM. The HMM can then be used to generate a sequence of observation symbols denoted by:

$$O = O_1\ O_2\ O_3 \cdots\ O_T. \tag{4.24}$$

where each observation O_t is one of the symbols from V (see item 2 in §4.3.2.4), and T is the number of observations in the sequence). The procedure for generating the observation sequence is as shown in Figure 4.6.

Given the form of HMM discussed in §4.3.2.4 and the observation sequence O generated from the model using the procedure shown in Figure 4.6, there are three basic problems of interest that must be solved for the model to be useful in real-world applications. In the following paragraphs, these three problems are posed. A discussion on the implications of these problems (as they relate to the problem of automatic speech recognition) is also offered.

Problem I:

Given the observation sequence $O = O_1\ O_2\ O_3 \cdots\ O_T$, and a model $\Lambda = (A, B, \pi)$, how does one compute $P(O \mid \Lambda)$, the probability of the observation sequence, given the model?

Step 1 Initialization:

Set $t = 1$;
Choose an initial state $q_1 = S_i$ according to the initial state distribution π.

Step 2 Symbol Generation:

Choose $O_t = V_k$ according to the observation probability in state S_i, *i.e.* b_{ik}.

Step 3 State Transition:

Transit to a new state $q_{t+1} = S_j$ according to the state transition probability in state S_i, *i.e.* a_{ij}.

Step 4 Time Increment:

Set $t = t + 1$.

Step 5 Termination Test::

Return to step 2 if $t \leq T$; otherwise terminate the procedure.

Figure 4.6: *Steps involved in generating an observation sequence from a completely specified HMM.*

Problem II:

Given the observation sequence $O = O_1\ O_2\ O_3 \cdots\ O_T$, and a model $\Lambda = (A, B, \pi)$, how does one choose a corresponding state sequence $Q = q_1\ q_2\ q_3\ \cdots\ q_T$ which is optimal in some meaningful sense (*i.e.* best 'explains' the observations)?

Problem III:

How does one adjust the model parameters $\Lambda = (A, B, \pi)$ to maximize $P(O \mid \Lambda)$?

The first problem is the evaluation problem. In other words, given a model and a sequence of observations, how does one compute the probability that the observed sequence is produced by the model. A more useful viewpoint is to see the problem as one of *scoring* how well a given model matches a given observation sequence. This viewpoint is important in cases where one has to choose among several competing models, because the solution to Problem I allows one to choose the model which best matches the observations.

The second problem is one in which one attempts to uncover the hidden part of the model, *i.e.* the state sequence. This is a typical estimation problem. An optimality criterion is usually imposed to solve this problem as well as possible. There are several possible optimality criteria that can be imposed and hence the choice of criterion is a strong function of the intended use for the uncovered state sequence. A typical use of the recovered state sequence is to learn about the structure of the model, and to get average statistics, behaviour, *et cetera*, within individual states.

The third problem is the one in which one adjusts (in an optimal manner) the model parameters so as to best describe how the observed sequence comes into being. The observed sequence in this case is called the training sequence since it is used to train the model. The solution to Problem III is the crucial one for most applications of hidden Markov models since it allows us to optimally adapt model parameters to observed training data, *i.e.* to create the best model for real observed (physical) phenomena.

4.3.2.6 Solutions to the three basic problems for HMM

Solution To Problem I:

The most straightforward way of computing the probability of the observation sequence, given the model is given by

$$P(O \mid \Lambda) = \sum_{q_1\ q_2\ q_3 \cdots\ q_T} \pi_{q_1} b_{q_1 O_1} \cdot a_{q_1 q_2} b_{q_2 O_2} \cdots a_{q_{T-1} q_T} b_{q_T O_T} \tag{4.25}$$

The interpretation of the computation in the above equation now follows. Initially (at time $t = 1$), we are in state q_1 with probability π_{q_1}, *and* generate the symbol O_1 with probability $b_{q_1 O_1}$. A transition is then made from state q_1 to q_2 with probability

Step 1 Definition:

The forward variable, $\alpha_t(i)$, is defined as

$$\alpha_t(i) = P(O_1, O_2, \cdots, O_t, q_t = S_i \mid \Lambda)$$

i.e. the probability of the partial observation sequence (until time t) and state S_i at time t, given the model Λ.

Step 2 Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N;$$

Step 3 Recursion:

for $t = 1, 2, \cdots, T-1, \ , 1 \leq j \leq N$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1})$$

Step 4 Termination:

$$P(O \mid \Lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Figure 4.7: *The forward algorithm for evaluating $P(O \mid \Lambda)$ efficiently.*

$a_{q_1 q_2}$, *and* generate symbol O_2 with probability $b_{q_2 O_2}$. This process continues until the last transition from state q_{T-1} to q_T is made with probability $a_{q_{T-1} q_T}$, *and* generate symbol O_T with probability $b_{q_T O_T}$.

The computation of $P(O \mid \Lambda)$, according to Equation (4.25) involves in the order of $2T \cdot N^T$ calculations, since at every time $t = 1, 2, \cdots, T$, there are N possible states to go through and for each summand, we need about $2T$ multiplications and additions (Rabiner and Juang, 1986). This calculation is computationally unfeasible, even for small values of N and T. For example for $N = 5, T = 100$, which is typical in isolated word recognition problem, there are of the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations! A more efficient procedure for solving Problem 1 is the forward algorithm (Baum *et al.*, 1970) as depicted in Figure 4.7.

From Figure 4.7, an order of $N^2 T$ additive and multiplicative operations is required for the computation of $\alpha_t(i)$. For $N = 5, T = 100$, this means about 3000 operations, in contrast with 10^{72} operations needed for the direct calculation. This is a savings of about 69 orders of magnitude.

Another efficient way of evaluating $P(O \mid \Lambda)$ in Equation (4.25) is the backward algorithm (Baum *et al.*, 1970). The steps involved in the backward algorithm is

Step 1 Definition:

The backward variable, $\beta_t(i)$, is defined as

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots, O_T \mid q_t = S_i, \Lambda)$$

i.e. the probability of the partial observation sequence from $t+1$ to the end, given state S_i at time t and the model Λ.

Step 2 Initialization:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N;$$

Step 3 Recursion:

for $t = T-1, T-2, \cdots, 1, \ , 1 \leq i \leq N$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

Step 4 Termination:

$$P(O \mid \Lambda) = \sum_{i=1}^{N} \beta_1(i)$$

Figure 4.8: *The backward algorithm for evaluating $P(O \mid \Lambda)$ efficiently.*

depicted in Figure 4.8. Like the forward algorithm, the computation of $\beta_t(i), 1 \leq t \leq T, 1 \leq i \leq N$, requires in the order of N^2T additive and multiplicative operations and can be implemented efficiently in a lattice structure (Rabiner, 1989).

The forward and backward algorithms (as depicted in Figure 4.7 and Figure 4.8 respectively) can also be combined to evaluate $P(O \mid \Lambda)$ according to

$$P(O \mid \Lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j) \tag{4.26}$$

for any t such that $1 < t < T-1$ (Ainsworth, 1988). In addition, the computations of the forward and backward variables, denoted by $\alpha_t(i)$ and $\beta_t(i)$ respectively, are used extensively in the solutions to Problems II and III.

Solution To Problem II:

There are several possible ways of solving Problem II, namely finding the "optimal" state sequence associated with the given observation sequence. The desired solution depends on the optimality criterion used. For example, one possible optimality criterion is to choose the states q_t which are *individually* most likely at any given time

t. Thus,

$$q_t = \text{argmax}[\gamma_t(i)], \quad 1 \leq t \leq T, \quad 1 \leq i \leq N. \tag{4.27}$$

where

$$\text{argmax}[\gamma_t(i)] = i \text{ iff } \gamma_t(i) \geq \gamma_t(j), i \neq j, 1 \leq i, j \leq N. \tag{4.28}$$

and

$$\gamma_t(i) = P(q_t = S_i \mid O, \Lambda) \quad 1 \leq t \leq T. \tag{4.29}$$

In Equation (4.29), $\gamma_t(i)$ is the probability of being at state S_i at time t, given the observation sequence and the model. Equation (4.29) can be evaluated in terms of the forward and backward variables, that is:

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)} \tag{4.30}$$

The normalization factor $\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)$ makes $\gamma_t(i)$ a probability measure so that

$$\sum_{i=1}^{N} \gamma_t(i) = 1 \tag{4.31}$$

While this criterion maximizes the expected number of correct individual states, there could be some problems with the resulting state sequence. For example, when the HMM has forbidden state transitions (*i.e.* $a_{ij} = 0$ for some i and j), the "optimal" state sequence may, in fact, not even be a valid state sequence. This is due to the fact that the solution simply determines the most likely state at every instant, *without* regard to the probability of occurrence of the sequence of states.

The above problem may be overcome by using a more suitable optimality criterion. For example, one could solve for the state sequence that maximizes the expected number of correct pairs of states, or triples of states, and so on. The most widely used optimality criterion is to find the *single* best state sequence (path), *i.e.* to maximize $P(Q \mid O, \Lambda)$ which is equivalent to maximizing $P(Q, O \mid \Lambda)$. A dynamic programming technique for solving this optimal path is called the Viterbi algorithm (Forney, Jr., 1973). The steps involved are shown in Figure 4.9. Again, this algorithm can be implemented efficiently using a lattice structure. As described in Chapter 5, the Viterbi algorithm is used for the training of the speech recogniser (which is based on the HMM) that I have implemented. The same algorithm is also used for the testing of the recogniser (see Chapter 5).

Solution To Problem III:

The problem of estimating the model parameter (so that the probability of the observation sequence given the model is maximized) is the most difficult of all the three problems that have been posed. To date, no *analytical* solution has been found for solving this problem directly. As a matter of fact, given any finite observation sequence as training data, there is no optimal way of estimating the model parameters (Rabiner, 1989). However, there are procedures which *iteratively* improves the estimates of HMM parameters $\Lambda = (A, B, \pi)$, so that $P(O \mid \Lambda)$ increases on every updated estimate of the parameters. The Baum-Welch method (Baum *et al.*, 1970), the

Step 1 Definition:

A new quantity, $\delta_t(i)$, is defined as

$$\delta_t(i) = \max_{q_1,q_2,\cdots q_{t-1}} P(q_1, q_2, \cdots, q_t = S_i, O_1, O_1, \cdots, O_t \mid \Lambda)$$

i.e. the probability of the most probable path, at time t, which accounts for the first t observations and ends in state S_i

Step 2 Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N;$$

$$\phi_1(i) = 0$$

Step 3 Recursion:

For $t = 2, 3, \cdots, T$ and $j = 1, 2, \cdots, N$

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \, b_j(O_t)$$

$$\phi_t(j) = \text{argmax}[\delta_{t-1}(i) a_{ij}]$$

Step 4 Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \text{argmax}[\delta_T(i)]$$

Step 5 Backtracking:

For $t = T-1, T-2, \cdots, 1$

$$q_t^* = \phi_{t+1}(q_{t+1}^*)$$

Figure 4.9: *The Viterbi algorithm for maximising* $P(Q \mid O, \Lambda)$

Step 1 Re-estimate π matrix:

$$\hat{\pi}_i = \gamma_1(i), \quad 1 \leq i \leq N$$

Step 2 Re-estimate A matrix:

$$\hat{a_{ij}} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \cdot a_{ij} \cdot b_j O_{t+1}(j) \cdot \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Step 3 Re-estimate B matrix:

$$\hat{b_{jk}} = \frac{\sum_{t=1, O_t = V_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Figure 4.10: *The Baum-Welch algorithm for re-estimating HMM parameters.*

gradient search technique (Levinson *et al.*, 1983b) and simulated annealing technique (Paul, 1985) are examples. These first two methods of estimating the parameters of the models, are discussed in detail by Levinson *et al.* (1983b). In particular, they show that the estimation problem is a constrained optimisation problem and hence can be solved by the classical method of Langrange multipliers (Levinson *et al.*, 1983b). They also discuss the practical problems of computer implementation.

The steps involved in the Baum-Welch re-estimation procedure are shown in Figure 4.10. In Figure 4.10, the variables $\hat{\pi_i}, \hat{a_{ij}}$ and $\hat{b_{jk}}$, denotes the updated estimates of π_i, a_{ij} and b_{jk} respectively. It is worthwhile to point out that all the variables on the left hand side (such as $\gamma_t(i), \alpha_t(i), \beta_{t+1}(j)$) of the re-estimation algorithms have been defined and evaluated while solving Problems I and II.

The re-estimation formula for π_i, shown in Figure 4.10 as $\gamma_1(i)$, is simply the probability of being in state q_i at time $t = 1$. The re-estimation formula for a_{ij}(see Step 2 of Figure 4.10) is the ratio of the expected number of transitions from state q_i to q_j, to the expected number of transitions out of state q_j. Finally the re-estimation formula for b_{jk} is the ratio of the expected number of times of being in state j *and* observing symbol k divided by the number of times of being in state j. Notice that the summation for b_{jk} is from $t = 1$ to $t = T$.

4.3.2.7 Other types of HMM

So far, we have considered only a special case of discrete hidden Markov models, *i.e.* those with discrete number of states and symbols and with unconstrained state transitions. This type of model has the property that every a_{ij} coefficient is positive, that is

$$a_{ij} > 0; \quad 1 \leq i, j \leq N \tag{4.32}$$

where N is the number of states in the HMM.

In speech recognition applications, constrained hidden Markov models have been

found to better account for observed properties of the signal being modelled than the unconstrained model (Rabiner, 1989). An example of a constrained model has already been shown in Figure 4.4. The model as shown in Figure 4.4 is known as the left-to-right model (Jelinik, 1976). In the left-to-right model, the states are only allowed to proceed from left to right (*i.e.* to a state with higher index) or stay in the same state.

The common feature of all left-to-right HMMs is that the state transition coefficients have the property that

$$a_{ij} = 0; \quad j < i \tag{4.33}$$

which means that no transitions are allowed to states whose indices are lower than that of the current state. A further constraint on the left-to-right model is that the state sequence must begin at state 1 and end in state N. In order to ensure that large changes in indices does not occur, additional constraint of the form

$$a_{ij} = 0; \quad j > i + \triangle \tag{4.34}$$

is often used. In particular, for the example of Figure 4.4, the value of $\triangle$ is 1, *i.e.* no jumps of more than 1 state is allowed.

It should be clear that the imposition of the constraints of the left-to-right model, or those of the constrained jump model, has no effect on the re-estimation procedure. This is because any HMM parameter set to zero initially, will remain at zero throughout the re-estimation procedure (see Figure 4.10).

The discrete HMM, while easier to implement, is inadequate for more difficult speech recognition tasks such as the recognition of continuous speech. To overcome this, semi-continuous (Huang and Jack, 1988) and continuous (Juang, 1985) hidden Markov models have been devised. Work on HMMs with sub-states have also been reported (Austin and Fallside, 1988).

The main difference between the (semi-)continuous HMMs and the discrete HMMs is that the observations (which are actually continuous signals or vectors) are not quantized into discrete symbols in (semi-)continuous HMMs. The observation probabilities between the former and the latter are also described differently, in the former, by an observation probability matrix, $B = \{b_{jk}\}$, and in the latter, by a continuous probability density function of the form

$$b_j(x) = \sum_{k=1}^{M} c_{jk} \mathbf{N}(x, \mu_{jk}, \mathbf{U}_{jk})$$

where $\mathbf{N}$ denotes a multidimensional Gaussian density function of mean vector μ,$\mathbf{U}$ a covariance matrix and c_{jk} a weighting matrix.

4.3.2.8 Comments

Although the HMM approach has contributed greatly to recent advances in speech recognition, it has intrinsic limitations. One of these is the assumption that speech is a strictly (hidden) Markovian process. Another limitation is the assumption that successive observations (frames of speech) are independent, and therefore the probability

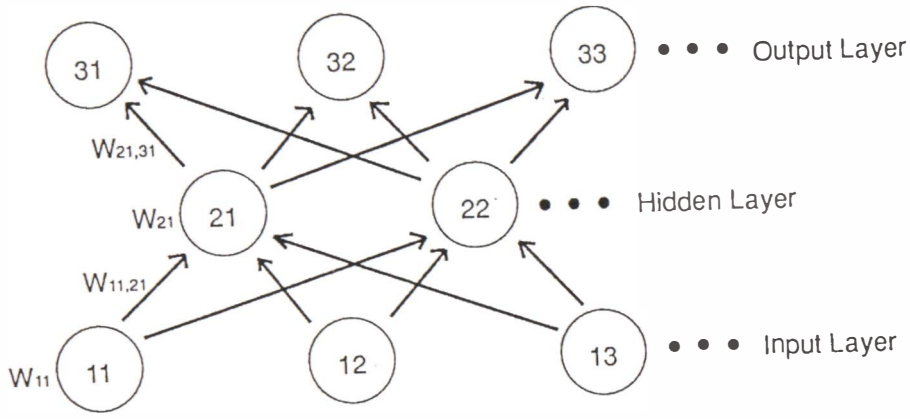


Figure 4.11: *A simple neural network. Each circle represent a neuron. At each neuron, the sum of all its weighted inputs are calculated. The sum is then passed through a nonlinear function to give its output. The output then serves as the input to the neuron in the next layer.*

of a sequence of observations can be written as a product of individual observations, *i.e.*

$$P(O_1, O_2, O_3, \cdots, O_T) = \prod_{t=1}^{T} P(O_i)$$

Finally, a first order HMM (which means that the probability of being in a given state at time t is only dependent on the state at time $t-1$) may not be appropriate for speech sounds where dependencies could extend through several states. However, in spite of these limitations, this type of statistical model has worked extremely for certain type of speech recognition problems. The fact that nearly all the currently commercially available speech recognisers are HMM-based (Levinson and Roe, 1990) is a testimony to the usefulness of this technique.

4.3.3 Neural network techniques

The application of neural network techniques to speech recognition problems is relatively recent compared to the DTW and HMM techniques. In the neural network approach, pattern classification is achieved with a multi-layer network of many nonlinear computational elements (see Figure 4.11). These computational elements or neurons are connected via weights that are typically adapted during training to improve performance. As shown in Figure 4.11, each neuron finds the sum of its

inputs, limits the range of its output with a nonlinear function, and passes the result to the neurons in the next layer. A commonly used nonlinear function is the sigmoid function (see Equation (2.19)). Each neuron in the output layer corresponds to a speech pattern which may be a word or a sub-word unit). When a speech pattern is presented to the input layer, the designated output neuron should respond.

Before a neural network can be used for speech recognition, its weights must be adjusted, *i.e.* it must be trained. One of the training algorithm is called the back-propagation technique (Lippmann, 1987). This involved presenting both the input pattern and the desired output pattern to the network, and the weights are then adjusted iteratively to minimize the difference between the actual output and the desired result.

Because neural networks are not as mature as the DTW and the HMM approaches, it is difficult to compare their relative merits (Levinson and Roe, 1990). At this time, successful implementation of neural network technique for speech recognition has been carried out by Waibel *et al.* (1989) and Kohonen (1988). The results of these two implementations are promising, but their superiority to the HMM technique has not been demonstrated.

Further discussion on the neural network technique is beyond the scope of this thesis. Readers who are interested in finding out more about neural networks are referred to the two tutorial papers by Lippmann (1987; 1988). The two special issues (September 1990 and October 1990) of the Proceedings of IEEE on neural networks should also be a valuable source of information.

4.4 Current capabilities of automatic speech recognition

Research in speech recognition has produced numerous commercial speech recognizers that can recognize words chosen from a small (generally consists of the digits, alphabets and/or a few command words) vocabulary spoken by almost any speaker over the telephone network (Picone 1990; Levinson and Roe 1990; Wallich 1987). In laboratories around the world, more advanced experimental speech recognition systems have been built. The Tangora system (Bahl *et al.*, 1988) and the SPHINX system (Lee *et al.*, 1990) are examples. The Tangora system operates in a speaker dependent mode. It can recognize up to 20000 isolated words. The SPHINX system, on the other hand, operates in a speaker-independent mode and can recognize continous speech constructed from a 1000-word vocabulary. Recognition rate in excess of 90% from these two experimental systems have been reported (Lee *et al.* 1990; Bahl *et al.* 1988). Table 4.2 summarises the recognition accuracy of several experimental speech recogniser. It is worthwhile to mention that all of the recogniser listed in Table 4.2 are HMM-based.

System	Task	Speakers	Style	Vocabulary	Word accuracy
SPHINX	DARPA	Independent	Connected	997	95.8%
Tangora	Typewriter	Dependent	Isolated	5,000	97.1%
Tangora	Typewriter	Dependent	Isolated	20,000	94.6%
AT&T	Phone No.	Independent	Connected	11	99.6%

Table 4.2: *Recognition accuracy of several experimental speech recogniser. Adapted from Mariani (1989).*

4.5 Summary

The automatic speech recognition problem has been viewed in terms of a pattern classification problem. Three different techniques for solving the automatic speech recognition problem have been outlined to varying degrees of detail. The basic concepts, strengths and weaknesses of dynamic time warping and hidden Markov modelling have been discussed in considerable depth. While the material on neural networks is rather cursory, this is compensated for (I hope) by the referral of relevant literature where more in-depth study of the technique has been documented. The chapter concludes with a review of the current status of speech recognition technologies.

Chapter 5

CONTRIBUTION TO COMPUTER SPEECH RECOGNITION USING A HIDDEN MARKOV MODEL

"Basic reseach is what I do when I don't know what I am doing."
(Wernher van Braun)

5.1 Introduction

The theory of the hidden Markov model (HMM) has been reviewed thoroughly in Chapter 4. In this Chapter, the implementation of a speech recogniser based on the HMM is discussed. The results of a series of recognition tests are also reported.

As evidenced by the literature cited in Chapter 4 and elsewhere in this thesis, the application of the HMM for speech recognition purposes has been extensively studied. From my literature study, it seems to me that in all of these HMM recognisers, the Baum-Welch algorithm (§4.3.2.6) has been used for the re-estimation of the HMM parameters and the Viterbi algorithm (§4.3.2.6) for scoring or evaluation purposes only (Rabiner *et al.*, 1989). In contrast, the speech recogniser described in this Chapter uses both the Viterbi algorithm for the training and the scoring or evaluation of the HMM recogniser. This new approach constitutes part of the original work reported in this thesis.

Furthermore, it appears to me that all the reported work uses analysis windows with both the length and the amount of overlap fixed. I have experimented with analysis window of fixed length but with "adjustable" amount of overlap, with the

result that all the words (that are used in training and testing) have the same number of frames. This seems a reasonable thing to do since it makes the scoring process "fairer" for all the words because each then has the same number of frames (of features). This is a novel approach and thus is also part of my original work. The results (see experiments 1 and 3 in Table 5.4 and the associated Table 5.5) verify the usefulness of this approach.

The organization of this Chapter is as follows. §5.2 provides an overview of the HMM speech recogniser that I have implemented. Two important issues regarding the implementation of the HMM are discussed in §5.3. The HMM speech recogniser is then evaluated on a database consisting of the digits zero to nine, spoken by a female New Zealand speaker. These experimental evaluations are described in §5.4, §5.5 and §5.7. Finally, a summary is provided in §5.8.

5.2 Overview

There are two phases involved in the recognition of an isolated word using HMM: the training phase and the classification (or recognition or testing) phase. In the training phase, the training set of observation sequences is used to derive a set of reference hidden Markov models, one for each word in the vocabulary. In the classification phase, the probability of generating the unknown observation sequence is computed for each reference model using the Viterbi algorithm (see §4.3.2.6). The unknown word is then classified as the word whose model gives the highest probability. Figure 5.1 shows a block diagram of the HMM recognizer. The details of each of the major components of the recognizer are explained in the following subsections.

5.2.1 Speech database

The speech database used in this experiment consists of 20 repetitions of the digits zero to nine uttered by a female New Zealander. The speech was recorded in a quiet computer room and then low pass filtered at 4.5kHz. This was sampled at 10kHz with 12 bits resolution. The endpoints of each utterance were then manually edited. The speech database was subsequently divided into two sets with the first ten repetitions of each digit being used as the training set and the next ten as the test set. The speech waveform of the word 'ONE' is shown in Figure 5.2 as an example.

5.2.2 Feature extraction

The digitized speech samples were each divided into frames of 200 samples (20ms) long with an overlap of 50 samples (5ms) between frames. Each frame was then windowed using a Hamming window before applying the Durbin-Levinson algorithm (see Figure 3.3) to produce a set of 10 linear predictive coefficients. Figure 5.3 shows a 20ms segment of the speech waveform of Figure 5.2. The resultant linear predictive coefficients extracted from the 20ms segment are shown in Figure 5.4.

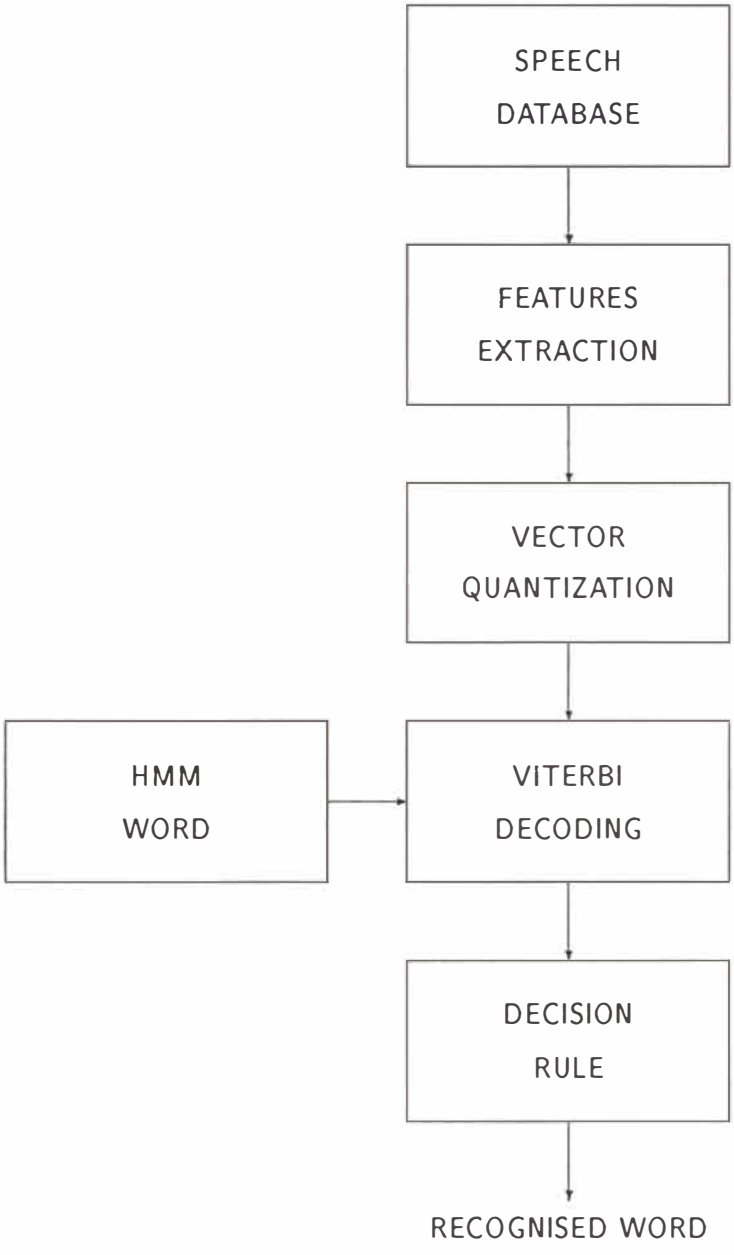


Figure 5.1: *Block diagram of the HMM isolated word recogniser that I have used to obtain the results described in §5.4.*

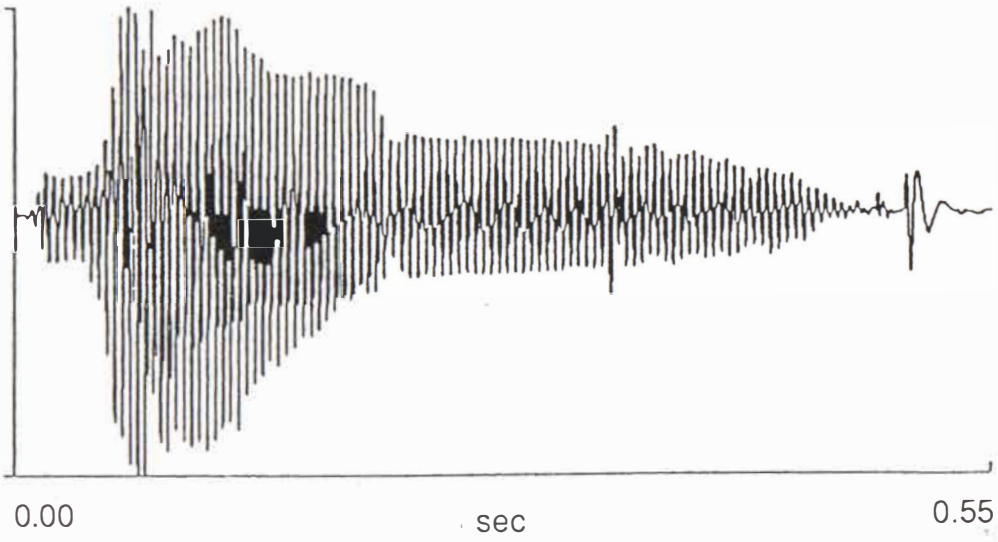


Figure 5.2: *Speech waveform of the word 'ONE'.*

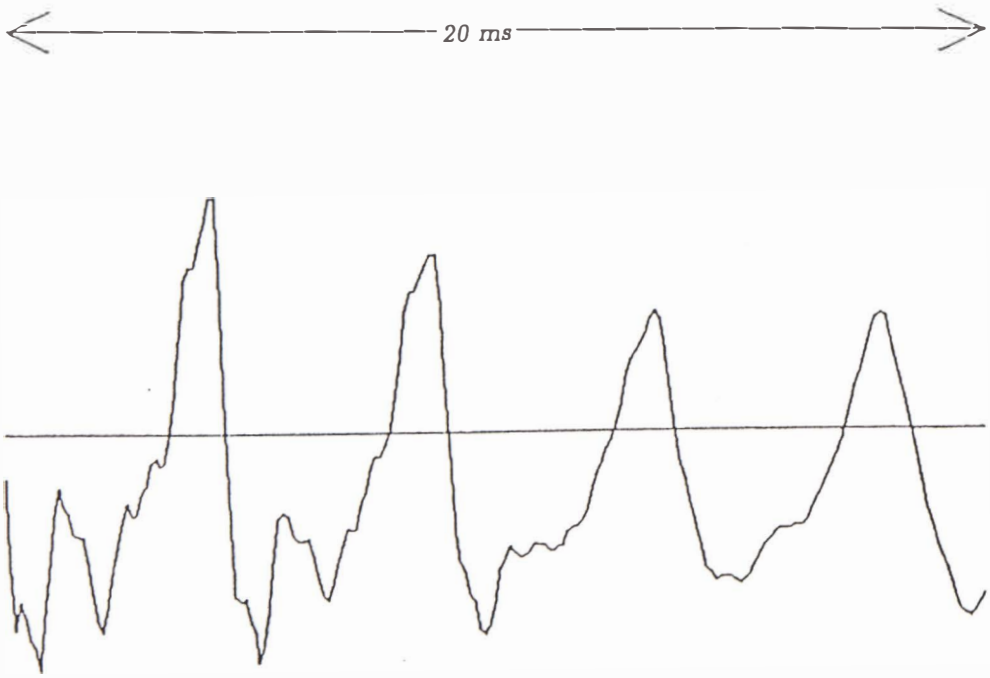


Figure 5.3: *A 20 ms segment of the speech waveform shown in Figure 5.2.*

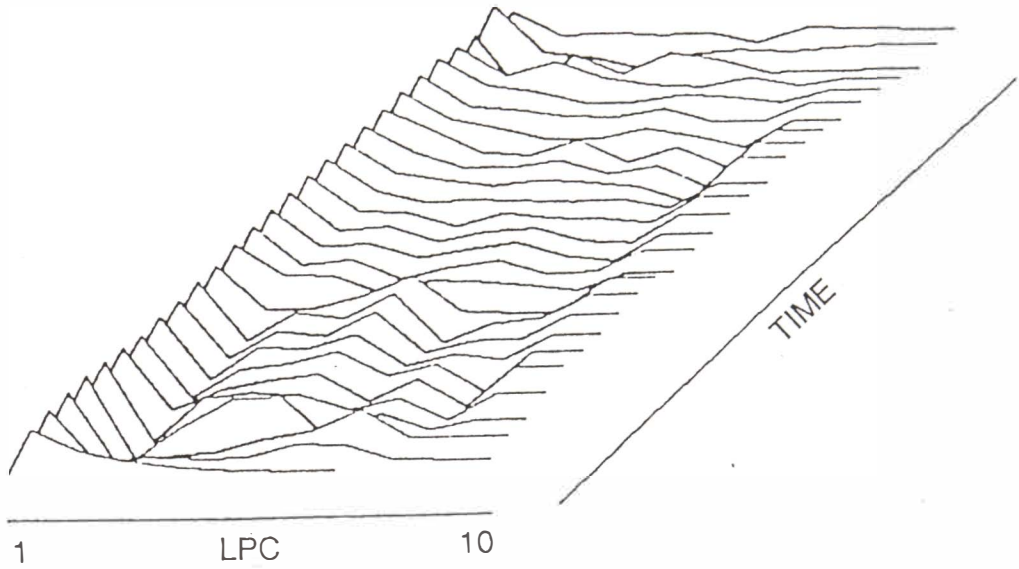


Figure 5.4: *The linear predictive coefficients of the speech waveform shown in Figure 5.2.*

5.2.3 Vector quantizer

The sets of LPCs derived from the training sets were then passed through a vector quantizer (see Chapter 2). The VQ processor effectively divided the 10-dimensional space of the LPCs into L (where L is a power of 2) regions, so that all the LPC vectors within each region were represented by the centroid of that region. These centroids (each a 10-D vector) are collectively known as the VQ codebook and each centroid is identified by an index. For example, a 10 dimensional VQ codebook of size 256 would contain 256 10-dimensional vectors, each identified by an index (or symbol). Thus, each index may have any integer value between 1 and 256. Thus, after vector quantization, a p-frame 10-dimensional LPC (10-LPC) becomes a 1-dimensional time sequence of length p. Figure 5.5 shows an example of a 7-frame 10-dimensional LPC vectors which has been quantized to 256 levels.

5.2.4 Training phase

During the training phase, a HMM is computed for each word of the digits zero to nine. This involves estimating, from a training set of vector-quantized 10-LPC features extracted from multiple utterances of each word, the complete set of HMM parameters, $\Lambda = (\pi, A, B)$.

The procedure for obtaining the estimates of the model parameters is shown in Figure 5.6. As shown in Figure 5.6, the first step in the training procedure is to decide on the configuration of the HMM and to choose an initial set of model parameters for

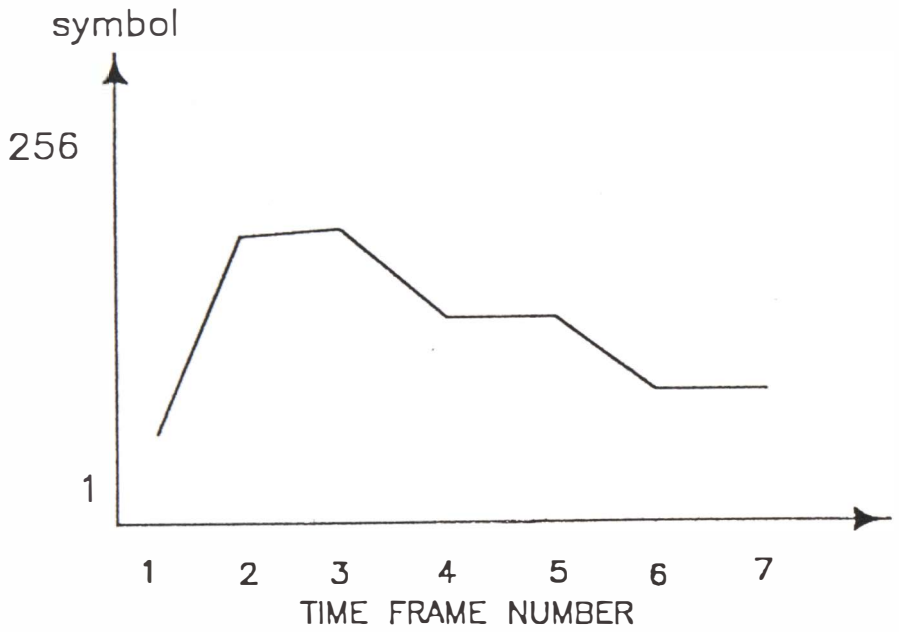


Figure 5.5: *An example of a 7-frame 10-dimensional LPC vectors which has been quantized to 256 levels.*

the selected HMM configuration. The configuration that I have chosen is shown in Figure 5.7. The HMM configuration shown in Figure 5.7 is known as the left-to-right configuration (§4.3.2.7) with the further constraint that no state transitions of more than 2 states is allowed, *i.e.*

$$a_{ij} = 0; \;\; j > i + 2 \tag{5.1}$$

Subject to the satisfaction of Equation (5.1) and all the probabilistic constraints which have already been discussed in §4.3.2, the initial values of the model parameters, $\Lambda = (\pi, A, B)$, are set randomly.

The second step in the training procedure is to segment each observation sequence into a state sequence. This segmentation is achieved by finding the optimum state sequence using the Viterbi (Forney, Jr., 1973) algorithm which has already been shown in Figure 4.9. From this decoded state sequence, a new model estimate is then derived as follows:

1. The new estimate of b_{jk}, denoted by $\hat{b}_{jk}$, is computed from

$$\hat{b}_{jk} = \alpha_{kj}/\beta_j \tag{5.2}$$

   where α_{kj} is the number of times symbol k is observed while in the jth state and β_j is the total number of times the model is in state j.

2. The new estimate of a_{ij} is obtained from

$$\hat{a}_{ij} = \gamma_{ij}/\eta_i \tag{5.3}$$

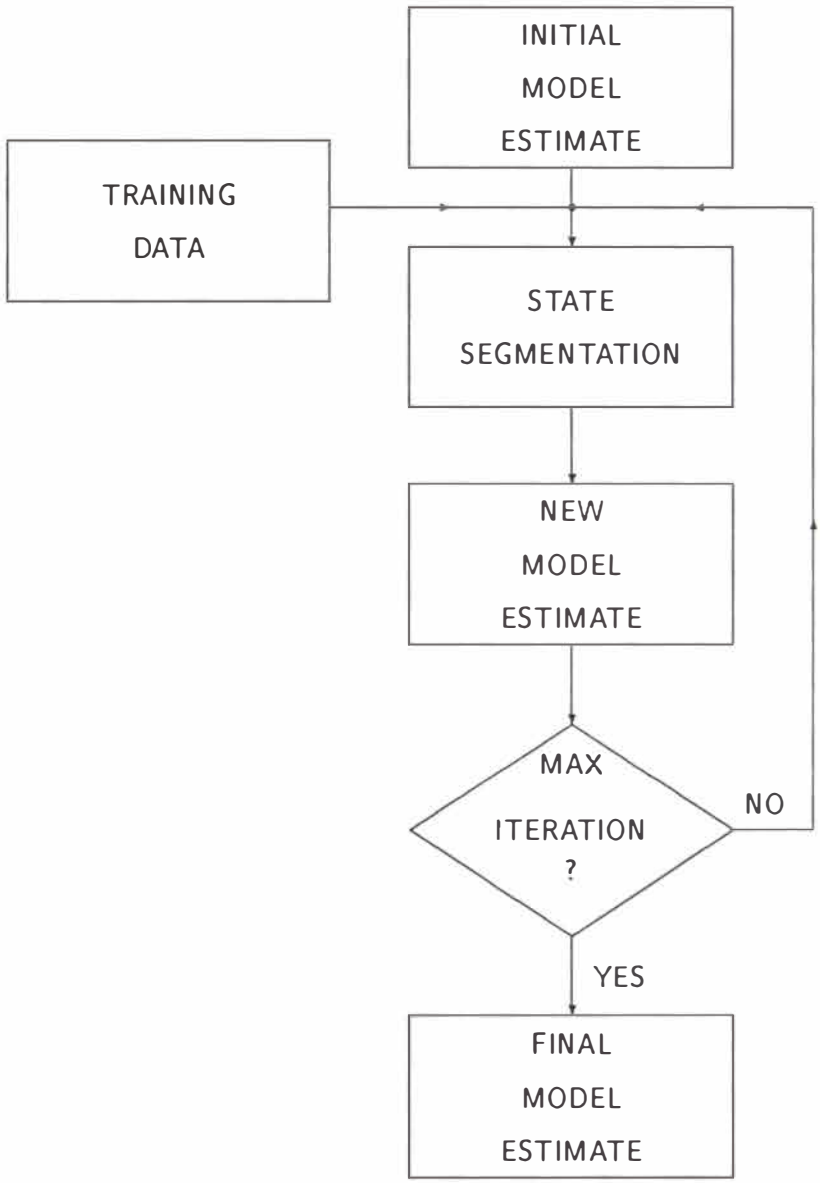


Figure 5.6: *Flow chart of HMM re-estimation.*

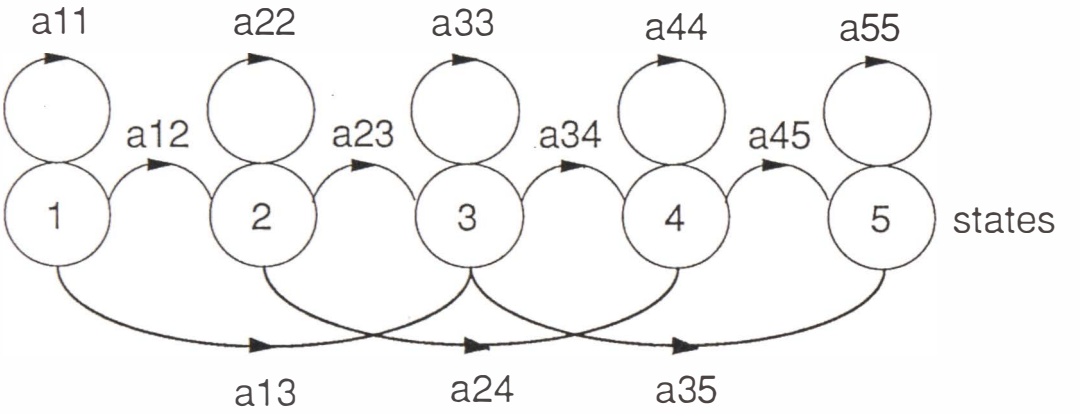


Figure 5.7: *Configuration of the HMM that I have used to obtain the results described in §5.4.*

where $\gamma_{i,j}$ is the number of transitions from state i to state j and η_i is the total number of transitions out of state i.

The iteration is repeated for a maximum of 10 times. This has been found to be sufficient for all my experiments.

Figure 5.8 and Figure 5.9 shows respectively, the final estimates of the state transition matrix, A, and the observation matrix, B for the word 'six'.

5.2.5 Testing/recognition phase

During the recognition phase, the digitized samples of a word from the unknown set is LPC-analysed the same way as before. This results in a sequence of 10-dimensional LPC vectors. The Euclidean distance between each of these vectors and the entries in the codebook is then calculated. Each vector is then assigned the index of the codebook entry closest to it. This results in a finite sequence of observations, $O = O_1, O_2, \cdots, O_T$, where O_t is the value of the assigned index at time t. Then, for each vocabulary word model, the most likely state sequence is found via the Viterbi algorithm and the log likelihood score for the optimal state sequence (or path), $\log P^*$, is computed. The procedure for computing P^* has already been depicted in Figure 4.9. The decision rule assigns the unknown word to the vocabulary word whose model has the highest log likelihood score.

$$\underline{\underline{A}} = \begin{bmatrix} 0.57 & 0.26 & 0.17 & 0.0 & 0.0 \\ 0.0 & 0.81 & 0.19 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.82 & 0.04 & 0.14 \\ 0.0 & 0.0 & 0.0 & 0.86 & 0.14 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Figure 5.8: *An estimated A-matrix for the word 'SIX'.*

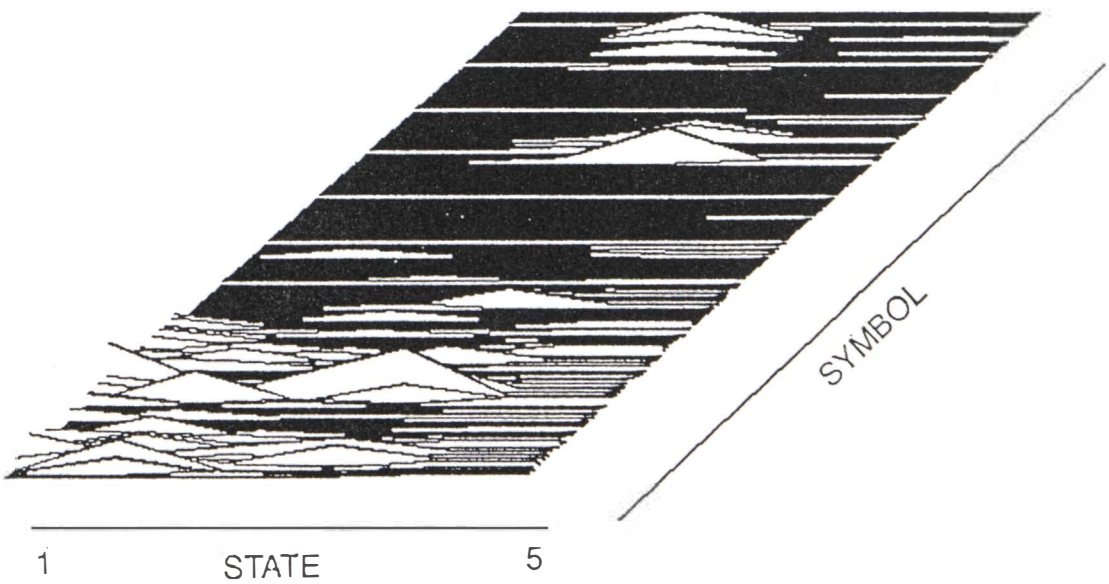


Figure 5.9: *An estimated B matrix for the word 'SIX'.*

5.2.6 Incorporation of state duration probability

As indicated by Equation (4.18), each state in the HMM has an *inherent* exponential duration probability function, $P_i(l)$. The state duration probability, $P_i(l)$, is the probability of staying in state i for l consecutive time frames. A state i, with a probability a_{ii} of returning to itself, has a state duration probability of

$$P_i(l) = (1 - a_{ii})a_{ii}^{l-1} \tag{5.4}$$

where l is the number of frames state i is occupied. From Equation (5.4), it can be seen that this exponential behaviour of the state duration probability is due to the presence of the self-transition coefficients, a_{ii}, $1 \leq i \leq N$.

Rabiner (1989) found that, for most physical signal, this inherently exponential state duration probability is inappropriate. Instead, it is better to explicitly model the state duration probability in some analytic form. This is achieved by setting to zero, all the self-transition coefficients a_{ii}, $1 \leq i \leq N$, and specifying an explicit duration probability function, $P_i(l)$ $1 \leq i \leq N$, for every state. For expedience and ease of implementation, the duration density is usually truncated at a maximum duration value L.

It should be clear that the HMM with explicit state duration probability can be made equivalent to the standard HMM by setting $P_i(l)$ to be the exponential density of Equation (5.4).

5.2.7 Re-estimation of state duration probability

With the explicit incorporation of the state duration probability into the formulation of a HMM, several minor changes must be made to the re-estimation formulas of the new set of HMM parameters. The derivations of these formulas are very similar to the derivations of the Baum-Welch algorithm (see Figure 4.10). The derivations can be found in Rabiner (1989) and are not repeated here.

While the incorporation of state duration probability improves the quality of the modelling of some problems, there are drawbacks to the use of this modified hidden Markov model. For example, the storage of $P_i(l)$ increases the storage load by L times, and the re-estimation of $P_i(l)$ increases the computational load by $L^2/2$ times (Rabiner, 1989). For many speech processing problems, L (which is the maximum duration value) is typically of the order of 25. This means that the computational load is increased by a factor of 300.

The state occupancy probability (defined above) can be derived using a much simpler procedure which make use of the state sequence decoded by the Viterbi algorithm. This procedure is described below (Rabiner, 1989).

Figure 5.10 shows an example of a segmented state sequence of a word. As shown in Figure 5.10, the state sequence is of T frames long, and state 3 is occupied for exactly l frames. By noting how long state 3 is occupied in *all* the segmented state sequences of the same word, we can compute $P_3(l/T)$, the probability of staying in state 3 for l/T of the time. The process is then repeated for all the other states and for every other word in the vocabulary. In my experiments, I have divided l/T into

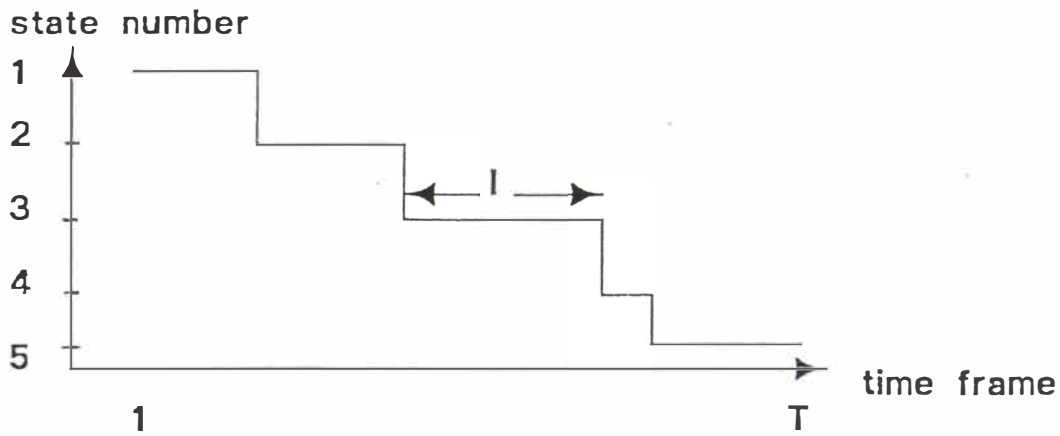


Figure 5.10: *Example of a segmented state sequence.*

10 discrete ranges $(0.0-0.1, 0.1-0.2, \cdots, 0.9-1.0)$, and the quantity $P_i(l/T)$ is estimated for each range.

During the recognition phase, the duration of each state is calculated from the decoded state sequence. The log likelihood, $\log P^*$, is then augmented by the log duration probabilities (suitably weighted by the duration weight, α) to give the final score for the recognition decision:

$$\log \hat{P} = \log P^* + \alpha \sum_{i=1}^{N} \log(P_i(l_i/T)) \tag{5.5}$$

where $\log P^*$ is the logarithm of the probability of observing the sequence O via the optimal state extracted using the Viterbi algorithm.

5.3 Implementation issues

5.3.1 Dealing with a finite amount of training data

The amount of training data available for estimating the HMM parameters is, of necessity, finite. If we look at the re-estimation formulas, we see that a parameter will be set to zero if there are no occurrences in the training set – *i.e.* if a symbol does not occur in the observation sequence, then the probability for that symbol will be zero in some states. If this effect is due to the small size of the training observation sequence, then special effort must be made to ensure that no HMM parameters becomes too small. If it is a real effect, then a zero probability is perfectly reasonable.

Vector Quantization Levels	Recognition Rate (%)
32	58
64	68
128	75
256	88

Table 5.1: *Recognition rate at various vector quantization levels.*

To deal with this problem, I have set a minimum value of 10^{-5} for all (except those transitions that are specifically forbidden) possible state transition and observation probabilities. This has been found to work satisfactorily.

5.3.2 Dealing with the local minima problem

While the re-estimation formulas improve the model estimate at each iteration, *i.e* $P(O \mid \hat{\Lambda}) > P(O \mid \Lambda)$ where $\hat{\Lambda}$ is the new estimate and Λ is the old estimate, the re-estimation may converges to a local maxima point rather than the global maxima. To overcome this problem, I repeat the re-estimation process 10 times, each time starting with a different initial model estimates which are set randomly. This results in 10 final estimates for each word, from which the N best (in terms of $P(O \mid \Lambda)$) estimates are chosen. For example, in the experiments described in §5.7, I have used the two best models for each word.

5.4 Experimental evaluations

Two sets of experiments were conducted, one to investigate the effects of different levels of vector quantization, the other to examine the effects of incorporation of state duration probabilities into the hidden Markov model.

5.4.1 Effects of different levels of vector quantization

To study the effects of different levels of vector quantization, a series of recognition tests were conducted using 32, 64, 128, and 256 levels of quantization. The results of these tests are listed in Table 5.1. The results clearly show that increasing the levels of vector quantization improves the performance of the HMM speech recognizer. Notice that the recognition rate increases from 58% when the quantization level was

Duration Weight α	Recognition Rate (%)
-1.0	64
-0.1	92
0.0	88
0.2	95
0.5	95
0.7	95
0.8	95
0.9	95
1.0	95
1.2	95
1.5	95
2.0	95
2.2	95
2.5	95
3.0	95
5.0	95
10.0	94
20.0	92
50	67

Table 5.2: *Recognition rate at various values of duration weight, α.*

at 32 to 88% when the quantization level was at 256. It is also noted that increasing the level of quantization to 512 resulted in no improvement to the recognition rate but doubled the amount of computational time.

5.4.2 Effects of state duration probability

To examine the effects of incorporating the state duration probability in the HMM model, I conducted a series of recognition tests by using the same set of experimental parameters as that described in §5.2.2 and varying the value of the duration weight α in Equation (5.5). The value of the duration weight α ranged between -1.0 and 50. It was found that, over this range, the recognition rate was above 90%, except at both the extremes. This, when compared to the recognition rate of 88% (see Table 5.1), represents a significant improvement. From Table 5.2, it is seen that the recognition rate was 95% for a wide range of the duration weight α. This indicates the robustness of the recognition system with respect to α.

No	Features	T_w(ms)	T_o(ms)	VQ Levels	Pre-emphasis	Accuracy(%)
1	10-lpc/rms	20	0	256	No	84
2	10-lpc	20	10	256	No	89
3	10-lpc	20	0	256	No	88
4	10-lpc	20	10	512	No	84
5	10-lpc	45	30	256	Yes	88
6	10-lpc	45	0	256	No	78
7	10-lpc	20	0	256	Yes	79
8	8-lpc	20	0	256	Yes	78
9	log(rms)	20	0	256	No	21
10	ZX	20	0	256	No	17

Table 5.3: *Accuracy of the HMM isolated digit recogniser using other features and under different conditions. Refer to the text for an explanation of the notations used in this table.*

5.5 Other features investigated

The results summarised in Table 5.1 and Table 5.2 have been obtained using the features discussed in §5.2.1. I have also studied the performance of the HMM speech recogniser using different features and under different conditions. Typical results of these studies are listed in Table 5.3. As shown in Table 5.3, I have used the 10th order LPC (10-lpc), the energy (rms), zero crossing rate of the speech waveform, the 8th-order LPC (8-lpc) and 10-lpc/rms (obtained by combining 10-lpc with rms to give a vector with 11 dimensions) as features for the HMM speech recogniser. Furthermore, I have used different frame lengths, T_w, and varying amount of overlap, T_o, between each frame. In addition, a raw speech waveform may or may not be pre-emphasised $(1 - 0.95z^{-1})$ before the features are extracted. The configurations of the HMM in all these tests is the same as before (see Figure 5.7).

5.6 Analysis and discussions of results

By studying the results in Table 5.3, the following observations can be made.

1. Combining rms with 10-lpc degrades the performance (by as much as 4% in the case of Experiments 1 and 3). This result is contradictory to reported work (see for example Rabiner and Wilpon (1987)). One may quite reasonably ask: Why did adding another feature (in this case, rms) degrade performance? What other ways can the rms and the 10-lpc (or any features at all) be combined so that the accuracy is improved? These questions should provide ample ground for future research.
2. Having an overlap gives a better result than having no overlap between frames (Experiments 2 and 3). This is in agreement with all the reported

work that I have read (Rabiner and Wilpon (1987, Pp349) for example). One plausible explanation for this is that by having overlap between frames, the movement (in time) of the relevant features (extracted within each frame) are better tracked. Again, why or how does a set of closely tracked (overlapping frames) features *actually* improves the recognition rate (over loosely tracked features)? These questions have not been addressed adequately and should be included in future studies.

3. A VQ level of 512 gives poorer results than a VQ level of 256 (see Experiments 2 and 4). This result is interesting because it indicates a reversal in trend indicated by the results which have already been shown in Table 5.1, where the recognition rate increases with increasing levels of vector quantisation. One may rightfully argue that since the higher the VQ level, the less (quantization) error is involved in the VQ process, one may expect a better result using a higher VQ level. The results (as indicated by Experiments 2 and 4 in Table 5.3) show the exact opposite is true.
   The decrease in accuracy (by increasing the VQ levels from 256 to 512) may be due to the continuous nature of the parameters (*i.e.* 10-lpc in the case of Experiments 2 and 4) themselves. Although it is possible to vector quantize these *continuous* parameters into *discrete* symbols chosen from a finite set of vectors, there may be serious degradation (in the sense that the recognition accuracy decreases) associated with such quantization beyond a certain *critical* VQ level. From what I have read and the results of my own experiments, the critical VQ level is at 256. Beyond this (VQ) level, these parameters (10-lpc) have been shown (Rabiner and Wilpon, 1987) to be better modelled by a (or a mixture of) multivariate continuous Gaussian distribution function, $N(f, \mu, U)$ where f is the parameters (or vectors) being modelled and μ, U are the mean and covariance matrix of the multivariate Gaussian distribution function.
4. Pre-emphasis does not necessarily improve the accuracy (Experiments 3 and 7). While pre-emphasizing the speech signals has been found to improve the recognition rates of automatic speech recognition experiments in most cases and has been more or less accepted as a necessary "signal conditioning" tool for speech processing (Ainsworth, 1988; Fallside and Woods, 1985; Witten, 1982) the results of Experiments 3 and 7 (see Table 5.3) show that pre-emphasis is *not always* a good idea. Of course, one can argue that by adjusting the configuration of the HMMs, one can eventually achieve a higher recognition rate with pre-emphasised signal, at the expense of a great deal of effort.
5. It is marginally better to use 10-lpc (Experiment 7) than 8-lpc (Experiment 8). This appears to agree with the results of Rabiner and Wilpon (1987) who have conducted more extensive experiments with 6th, 8th, 10th and 12th order LPC and concluded that: "In general, performance using a sixth order system is somewhat worse than that obtained using an eighth, 10th, or 12th order system. However differences in performance of the 8th, 10th, and 12th order systems are small and appear to be statistical in nature". Hence the optimum order (as long as it is

Number	Speaker	Features Set	Accuracy(%): 1 model	Accuracy(%): 2 models
1	DR	C	90	89
2	DR	A	98	98
3	DR	D	97	98
4	AE	A	90	86
5	AE	D	84	85
6	CA	C	97	97
7	CA	B	99	99

Table 5.4: *Accuracy of the HMM isolated digit recogniser with different speakers and one/two models for each word.*

at least of the 8th order) of LPC to use would be largely dependent on the computing power and the storage space of the machine on which the HMM speech recogniser is implemented.

6. The recognition results using the energy log(rms) (21% in Experiment 9) and the zero-crossing rate (17% in Experiment 10) of the speech signals are remarkable, in the sense that they are still better than mere guessing, which should give (on the average) 10% recognition rate. Furthermore, these two parameters are only one-dimensional (as compared to 10-lpc which is 10-dimensional). In addition, the calculation of these two parameters are straightforward and involve far fewer computations than say, the calculation of 10-lpc. The results of other researchers (Clark *et al.*, 1990; Rabiner and Levinson, 1981) show that these two parameters can be usefully combined with 10-lpc to improve the performance of a system using only 10-lpc.

5.7 Experiments with other speakers

Table 5.4 shows another set of evaluation tests that I have conducted. The difference between this set of tests and all the previous ones is that the isolated digits are uttered by three different speakers. As shown in Table 5.4, four different sets of features have been used. The descriptions of these four sets of features, which are identified by the letters A, B, C and D, are listed in Table 5.5. The HMM configurations as shown in Figure 5.7 have been used for all the tests described here. Notice that this is the same HMM configuration that has been used in all the previous tests. A VQ level of 256 has been used throughout. The same training procedures (as described in §5.2.4) have been used for all the tests.

From Table 5.4, it can be concluded that, regardless of the individual speaker and the set of features that have been used, no consistent improvement can be gained

A	10th order reflective coefficients, no pre-emphasis, 20 ms Hamming window with 5 ms overlap.
B	8th order reflective coefficients, pre-emphasis $(1 - 0.95z^{-1})$, 20 ms Hamming window with 5 ms overlap.
C	10th order predictive coefficients, no pre-emphasis, 20 ms Hamming window with 5 ms overlap.
D	10th order reflective coefficients, no pre-emphasis, 20 ms Hamming window with adjustable overlap so as to give a total of 64 frames for each word.

Table 5.5: *Descriptions of the four sets of features identified by the letters* A, B, C, D *in Table 5.4.*

by having two (instead of one) reference hidden Markov models for each word of the vocabulary. The main drawback of using two reference HMMs for each word is that, for each unknown word, the number of comparisons required in order to identify the unknown word is doubled. Therefore, the use of two models for each word is unjustified.

An interesting observation can be made by comparing the results of, say, Experiments 2 with 4, 3 with 5 and 1 with 6 9refer to Table 5.4) where the same sets of features have been used in each pair of experiments. These comparisons lead me to conclude that the accuracy of the HMM speech recogniser is highly speaker-dependent (given that the same amount of training time and the same training procedure is used for each speaker). This may be due to the accent or the manner of speaking of the particular speaker. Of course, with greater amount of training and experimenting (by varying systematically each parameters of the HMM in turn), one can *eventually* achieve "good" recognition results for any speaker with any accent.

It has also been found that, for all these speakers, the recognition rates can be improved by incorporating the state duration probability and by choosing the correct duration weight, α. It is unfortunate that there is no analytical way of deciding on the appropriate weight to use. Experimenting with different weights systematically seems to be the best way available at the moment.

5.8 Summary

I have implemented a discrete HMM speech recognition system in software and conducted a series of recognition tests to evaluate the performance of the HMM speech recogniser. The results of these experiments (which typify the other experiments that I have conducted) have led me to the following conclusions:

1. A vector quantization level of 256 is the optimum.
2. The incorporation of the state duration probability into the HMM can improve the recognition rate, provided a suitable duration weight, α (see Equation (5.5)), is applied to the state duration probability. In all my experiments, a weight between 1 and 5 has been found to be the most effective.
3. There is no analytical way of deciding on the appropriate weight to use. Experimenting with different weights systematically seems to be the best way available at the moment.
4. There is no advantage in using two reference models for each word instead of one.
5. The accuracy of the HMM speech recogniser varies with each individual speaker. This is very likely due to the variations in the manner of speaking of the particular speaker. Thus, the accuracy tends to be higher for someone who carefully enunciates each word.
6. Overlapping the frames from which the speech features are extracted leads to an improvement in recognition rate. The drawback is that the sequences of observations are longer as a result of frame-overlapping, thus the required processing time is correspondingly longer.
7. The results of other studies on speaker-independent isolated-word recognition with a 10-digit vocabulary have shown that the accuracies of HMM speech recognisers vary between 97.1% and 98.1% depending on the precise implementation details (Grant, 1991, Pp 48). Hence, the results reported in this chapter are comparable to those of other similar studies.

Chapter 6

CONTRIBUTION TO SPEECH CODING

"It is a good morning exercise for a research scientist to discard a pet hypothesis every day before breakfast. It keeps him young."
(Konrad Lorenz, On Agression)

A speech coding scheme devised by the speech group of the University of Canterbury is investigated, modified and evaluated. The theoretical basis of this scheme, henceforth referred to as the SAA/CLEAN coding scheme, lies in the now well-established source-filter model of speech production (see §1.6) where a segment of (voiced) speech is modelled as a result of the *convolution* between a train of *glottal pulses* and the *vocal tract filter*.

The SAA/CLEAN coding scheme uses shift-and-add (SAA) technique (see Thorpe (1990, Chapter 4) and §2.2.6), to extract the *average glottal pulse* or the SAA signal of the speaker. This average glottal pulse is then used as the kernel in a *subtractive deconvolution* scheme called CLEAN (see Thorpe (1990, Chapter 5) and §2.2.7). The result of the deconvolution is a CLEAN signal which can be interpreted as an estimate of the vocal tract filter response (Thorpe, 1990).

As it turns out, the CLEAN signal consists of only a few "non-zero" pulses. Since only the glottal pulse and the non-zero values of the sparse CLEAN signal need to be coded, a reduction in the number of bits required to encode the speech can be achieved. However, it has been found that the speech encoded using the original SAA/CLEAN scheme (Thorpe, 1990), while highly intelligible, contains unpleasant "click" sounds. The original coding scheme is modified by adding a "top-hat" to the average glottal pulse prior to the CLEAN deconvolution. This has been found to be effective in removing the unpleasant "click" sounds.

Three different shapes of top-hat with varying heights have been systematically studied. The performance of this modified SAA/CLEAN coding scheme at 16 kbits, 8 kbits and 4 kbits per second has been evaluated subjectively using the Mean Opinion Score (MOS) technique (IEEE, 1969). The evaluations have been carried out in a language laboratory by 49 human subjects. These assessments reveal a MOS of 3.33, 2.83 and 2.09 at bit rates of 16, 8 and 4 kilo bits per second (kbps) respectively.

The organization of this Chapter is as follows. All the necessary preliminaries required to explain the original SAA/CLEAN speech coding scheme are set out in §6.1. The modified SAA/CLEAN speech coding scheme is then outlined in §6.2. The experiments that have been performed are documented in §6.3 and the results are discussed in §6.4. Finally, §6.5 summarises the chapter.

6.1 The SAA/CLEAN speech coding scheme

Shift-and-Add (SAA) and CLEAN were originally devised for use with astronomical data but the techniques are proving useful in other fields, notably ultrasonic imaging and speech processing (Thorpe, 1990). The background and the wide ranging applications of these two techniques have already been detailed in the forms of published papers in learned journals (Högbom, 1974; Bates and Robinson, 1981), published book (Bates and McDonnell, 1989) and unpublished thesis (Davey, 1989; Thorpe, 1990).

The procedures for SAA and CLEAN operation have been described earlier in §2.2.6 and §2.2.7 in the context of speech analysis techniques. By combining these two techniques, Thorpe (1990) has devised a novel speech coding scheme and successfully demonstrated the viability of the scheme. This novel speech coding scheme (which is referred to as the SAA/CLEAN speech coding scheme) is described in the following subsections where relevant terminology is introduced.

6.1.1 The shift-and-add (SAA) algorithm

Shift-and-add (SAA) is a very powerful technique for removing distortions or blurrings from an ensemble of images/signals of an object of interest which has been distorted or blurred differently. The technique consists of shifting each blurred image/signal until its point of greatest amplitude is at the origin. All the blurred images/signals are then added. This has the effect of retaining the unchanging part of the images/signals (which, hopefully, is the same as the object of interest) while blurring out the variable part and the noise. In the context of speech processing, the unchanging part of the speech signal corresponds to the glottal excitation while the changing part corresponds to the time varying vocal-tract filter response (Thorpe, 1990).

6.1.1.1 SAA as an effective average glottal pulse extractor

The "source-filter" model (introduced in §1.6) of speech production is shown in Figure 6.1. In this model, a speech signal is regarded as the outcome of apply-

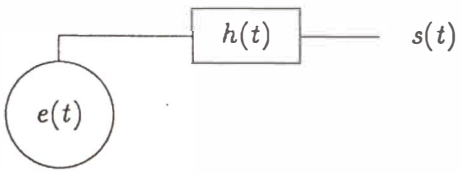


Figure 6.1: *The source filter model for human speech production.*

ing an excitation $e(t)$ to the vocal tract, $v(t)$ which is modelled as a time varying filter. During voiced speech (§1.4), the excitation consists of a sequence of pressure pulses emitted from the glottis, while during unvoiced speech (§1.4), the excitation consists of random noise excitation caused by air turbulence in the vocal tract.

For clarity reason and ease of explanation, I shall first deal with voiced speech only, and defer the discussion on unvoiced speech until §6.1.4.4.

A period (typically 10ms for male and 5ms for female) of voiced speech can be modelled by

$$s(t) = g(t) \odot v(t) \tag{6.1}$$

where $\odot$ is the convolution operator, $g(t)$ the glottal pulse, and $v(t)$ the vocal tract response. I shall refer to Equation (6.1) as the ideal convolution model.

As voiced speech is nearly repetitive at the pitch frequency, it can be broken down into segments, each of about one pitch period in length, *i.e.*

$$s(t) = \sum_{m=1}^{M} s_m(t - T_m) + c(t) \tag{6.2}$$

$$s_m(t) = g_m(t) \odot v_m(t) \tag{6.3}$$

where T_m is the time at which the m^{th} speech segment occurs, $v_m(t)$ the vocal tract response and $g_m(t)$ the glottal pulse for the m^{th} speech segment. The variable $c(t)$ is included to accommodate contaminations, such as noise and imperfect alignment in the superposition process. Hence, the variable $c(t)$ represents deviations from the ideal convolution model as indicated by Equation (6.1).

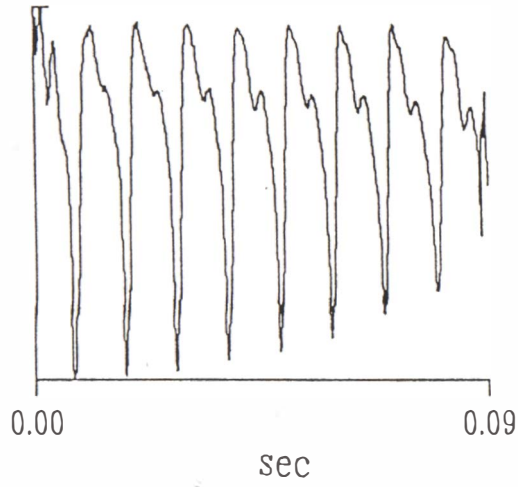


Figure 6.2: *A typical segment of voiced speech. Notice the periodic structure of the waveform.*

It is known that (Brieseman *et al.*, 1987) while the vocal tract changes shapes as different sounds are spoken, the glottal pulse remains relatively constant for a given speaker. Therefore, by applying SAA on the speech segments denoted by Equation (6.2) and Equation (6.3), an estimate of the average glottal pulse can be obtained. It should, however, be pointed out that the glottal pulse does change as different sounds are spoken, and especially as the pitch is varied. Therefore SAA cannot extract individual pulse shape but produces an average glottal pulse $\hat{g}(t)$, which is that part of the speech having a constant shape in each pitch period (Thorpe, 1990).

A typical segment of voiced speech is shown in Figure 6.2. SAA is then applied to about 300 pitch periods to produce a glottal pulse as shown in Figure 6.3.

6.1.1.2 Relevant comments

The following paragraphs examine the SAA algorithm as an effective nontactile means for the extraction of the *average* glottal pulse. These comments have been compiled largely from a recently completed PhD thesis of my learned colleague (Thorpe, 1990). These comments also applies to my own experience with the SAA processing of speech signal.

As Equation (6.2) indicates, SAA is performed on segments $s_m(t)$ of speech. Each segment contains a single pitch interval of voiced speech extracted from a complete utterance $s(t)$. One of the attractive aspects of SAA is that these segments do not need to be explicitly identified before applying the SAA algorithm. This is because

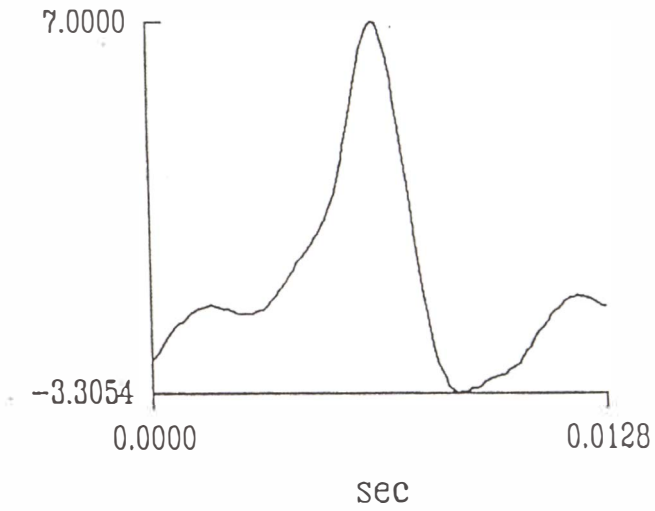


Figure 6.3: *The resultant estimated glottal pulse by performing SAA operation on a segment of speech, part of which is shown in Figure 6.2. Notice that the endpoints are non-zero. Compare this with Figure 6.4.*

the segments can be identified during the SAA algorithm itself, hence there is no need for accurate pitch or voiced/unvoiced analysis (Thorpe, 1990, Pp117).

The form of the SAA signal is insensitive to the length of each segment and the spacing between segments provided that the segment duration is no less than the average pitch period for the utterance being processed (Thorpe, 1990, §4.2.4.1). Because of this robustness of the SAA algorithm, it is computationally convenient to fix the length of the segment and the spacing between segments. The default values for these two values are 12.8 ms and 10.0 ms respectively. These values have been adopted in all the experiments reported herein.

The form of the SAA signal, however, is sensitive to the context and the duration of the utterances that have been used in the experiment. In order to obtain *reproducible* results *i.e.* consistent SAA signal, *phonetically balanced* (IEEE, 1969; Thorpe, 1990) sentences should be used. With regard to the duration of the utterance, computation experience suggests that 10 seconds is an adequate duration for a typical utterance. However, reliable SAA estimates can be obtained from utterances lasting no longer than 3 seconds, if they are carefully chosen to have appropriately balanced phonetic content (Thorpe, 1990, Pp 122-123).

It has also been observed (Thorpe, 1990, Pp123) that the differences between the SAA signals obtained from different phrases are more pronounced than that obtained from different occurrences of the same phrase. Furthermore, a SAA signal that is derived from an utterance that has been uttered in a relaxed manner is distinctively different from one which has been derived from an utterance that has been uttered in a tensed manner.

6.1.2 The CLEAN algorithm

As intimated in §2.2.7, CLEAN (Bates and McDonnell, 1989, pp79–84) is an iterative *subtractive deconvolution* techniques which was originally developed for deblurring astronomical images (Högbom, 1974), and has since been applied to ultrasonic imaging (Bates and Robinson, 1981) and many other applications (Bates and McDonnell, 1989). A more comprehensive review on the historical background of the CLEAN algorithm may be found in Thorpe (1990).

The CLEAN algorithm, as it applies to speech processing, consists of the following iterative steps (Thorpe, 1990), which are repeated until the number of pulses added to the CLEAN signal reaches some limit N_p, or until the maximum value of the residual signal, $r_i(n)$, at the i^{th} iteration, is less than some threshold *thres*, or until the maximum allowable number of iterations, i_{max}, is reached.

It should be mentioned that, the 5-step CLEAN process described below is applied to *segments* (see §6.1.4.1) of speech rather than a complete utterance.

1. Initialise the residual signal $r_0(n)$ by setting it to the speech signal $s(n)$:

$$r_0(n) = s(n) \tag{6.4}$$

2. Initialise the CLEAN signal $a_0(n)$ by setting it to the same length as the

speech signal, and filled with zeroes:

$$a_0(n) = 0 \tag{6.5}$$

3. For the i^{th} iteration, search through the residual signal $r_i(n)$ to locate the sample with the maximum (absolute) amplitude. Denote the position of this sample by k_i.
4. Update the CLEAN signal $a_i(n)$ by adding a pulse of amplitude f_i at position k_i:

$$f_i = r_i(k_i) \times \frac{\alpha}{g_{max}} \tag{6.6}$$

$$a_{i+1}(k_i) = a_i(k_i) + f_i \tag{6.7}$$

where α is the loop gain factor, and the maximum value of the SAA signal is denoted by g_{max}. Note that the choice of the loop gain factor affects both the speed of convergence to a solution and the form of the final CLEAN signal (Thorpe, 1990, Pp155–158).

5. The *kernel*, $g(n)$, is scaled by f_i and subtracted from $r_i(n)$ at the position k_i to form the new residual:

$$r_{i+1}(n) = r_i - f_i g(n - k_i) \tag{6.8}$$

It is appropriate to introduce, at this point, the term "CLEAN pulses" to denote the set of non-zero samples of the CLEAN signal. Mathematically, the CLEAN pulses can be expressed as the following set of number pairs

$$\{k_i, f_i); \quad i = 1, 2, \cdots N_p\} \tag{6.9}$$

where k_i, f_i are respectively, the position (in time) and the amplitude of each CLEAN pulse *i.e.* the non-zero sample of the CLEAN signal and N_p is the maximum allowable number of CLEAN pulses in each segment to be analysed.

6.1.3 Reconstructing speech from the CLEAN signal

Speech can be reconstructed from the CLEAN signal, $a(n)$, simply by convolving it with the kernel signal, $g(n)$. This is essentially the reverse of the CLEAN process outlined above (§6.1.2). The convolution can be computed very efficiently by taking advantage of the "sparcity" of the CLEAN signal, $a(n)$. The following sequence of steps represent one way of achieving this (Thorpe, 1990):

1. Initialise the reconstructed signal, $\hat{s}(n) = 0$.
2. For each CLEAN pulses, add to the reconstructed speech, $\hat{s}(n)$, the scaled (by f_i) version of the kernel, $g(n)$, at location k_i. That is,

$$\hat{s}(n) = \hat{s}(n) + f_i g(n - k_i) \tag{6.10}$$

6.1.4 Implementation details

I have described, in §6.1.1, §6.1.2 and §6.1.3 respectively, the general outline of the SAA algorithm, the CLEAN algorithm and how these two algorithms can be combined together to form a novel speech coding technique. I shall described below, a few implementation details of the SAA/CLEAN coding algorithms.

6.1.4.1 Segmentation considerations

As mentioned earlier, a speech utterance is divided into *segments* before the CLEAN operation is performed (on each segment). The total CLEAN signal is then formed by concatenating the individual CLEAN signals from each segment. However, arbitrary segmentation of the speech segment can leads to "*segmentation edge effects*" (Thorpe, 1990, P p158). These effects (described below) can be minimised by taking appropriate care in the segmentation process.

Segmentation edge effects are caused by the finite duration of the CLEAN kernel, $g(t)$, and the truncation inherent in the segmentation process (Thorpe, 1990, P p160). Thus, when a CLEAN pulse occurs at the end of the k^{th} segment, the shifted kernel (step 5 of the CLEAN algorithm in §6.1.2) extends beyond the end of the segment into the $(k+1)^{th}$ segment. This problem can be solved by modifying the $(k+1)^{th}$ segment. The modification is effected by subtracting out from the *residual* signal, the part of the shifted kernel which extends into the $(k+1)^{th}$ segment, before the CLEAN operation is performed on this segment.

A similar edge effect occurs when a CLEAN pulse occurs at the start of a segment. As a result, the shifted kernel extends into the *previous*, $(k-1)^{th}$, segment. If these was no segmentation, the subtraction of these copies of the kernel may cause additional pulses to be identified by subsequent iterations of the CLEAN algorithm in that part of the signal corresponding to the previous segment. However, because of segmentation, the CLEAN signal in the previous segment cannot be updated when the kernel extends into the previous segment. As a result, existing CLEAN pulses near the end of the $(k-1)^{th}$ segment may have incorrect amplitudes, and pulses that would have existed in the absence of segmentation may not be present.

Another type of segmentation edge effect occurs when the location of the sample with the maximum (absolute) amplitude is located at the end of the segment being processed, but when that sample is not a local extremum of the speech utterance. This results in an erroneously positioned peak, which can be avoided by testing for the possibility of such an occurrence in step 3 of the CLEAN algorithm as described in §6.1.2.

Because of the edge effects described above, it seems that the duration of each segment should be as large as is convenient. However, larger segments require more computation, since each iteration of the CLEAN procedure must search through more samples before it locates the maximum. In order to partially allay the effects of segmentation, adjacent segments are overlapped, so that the regions in which edge effects occur (*i.e.* approximately half the duration of $g(t)$ from each end of the segment) are processed in both the k^{th} and the $(k+1)^{th}$ segments. In the experiments described in §6.3, the duration of each segment was set at 128 samples

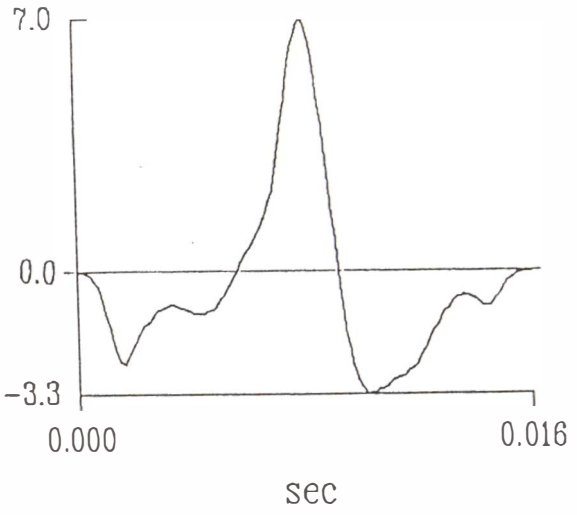


Figure 6.4: *The edge-extended glottal pulses. Notice that the endpoints of the glottal pulse have zero values. Compare this with Figure 6.3.*

(which corresponds to 12.8ms at a sampling rate of 10kHz), while the spacing between segments are set at 100 samples (10ms), giving an overlap of 28 samples (2.8ms).

6.1.4.2 Necessary modification to the glottal pulse

The glottal pulse extracted using the SAA method will usually have non-zero values at the starting and ending points, as shown in Figure 6.3. This is undesirable as it introduces *discontinuities* in the reconstructed speech.

To avoid the discontinuities problem, the glottal pulse is modified so that the starting and the trailing edges approach zero smoothly, as shown in Figure 6.4. This modification process is known as edge-extension and is achieved by dividing a 4-term Blackman-Harris window (Harris, 1978) into two halves and then attaching each half (suitably scaled) to the starting and trailing edges of the glottal pulse.

6.1.4.3 Optimisation of the CLEAN pulses

In order to improve the quality of the reconstructed speech, the amplitudes of the pulses in the CLEAN signal, a_i, are optimised by minimising the squared error between the reconstructed speech, $\hat{s}(n)$, and the original speech, $s(n)$, where the reconstructed speech $\hat{s}(n)$ is given by

$$\hat{s}(n) = \sum_m g(n-m)a(n) \tag{6.11}$$

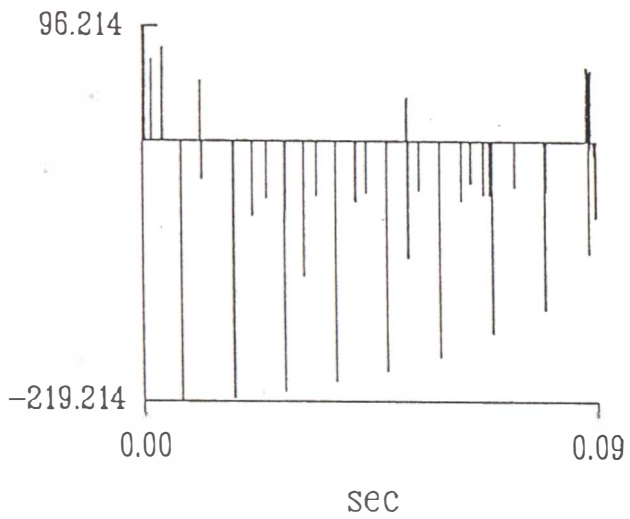


Figure 6.5: *Results of performing CLEAN operation on the speech signal shown in Figure 6.2, using the edge-extended SAA signal shown in Figure 6.4.*

Figure 6.5 shows the CLEAN signal of the speech segment shown in Figure 6.2, using the edge-extended SAA signal (Figure 6.4) as the kernel. The optimised version of the CLEAN signal appears in Figure 6.6.

The reconstructed speech obtained by convolving the CLEAN signal in Figure 6.6 and the SAA signal of Figure 6.4 is shown in Figure 6.7. Comparison of the reconstructed signal with the original speech in Figure 6.2 gives a 'visual' measure of the reconstruction 'quality'.

6.1.4.4 Dealing with unvoiced speech

From earlier discussions, it has been stated that the voiced and unvoiced sections of a speech signal are so different in character that they must be separated and processed independently by SAA. The same reasoning applies to CLEAN processing, since it makes no sense to deconvolve a signal that represents the excitation of the voiced sections of an utterance from a section of unvoiced speech.

The most straightforward way to deal with the differences between voiced and unvoiced sections of speech is, firstly, to separate them by means of voiced/unvoiced analysis techniques, and secondly, to perform SAA and CLEAN separately on the voiced and unvoiced sections. However, this is often inadequate, both because of errors in the voiced/unvoiced classification, and because of the occurrence of sections of speech which cannot be classified as voiced or unvoiced.

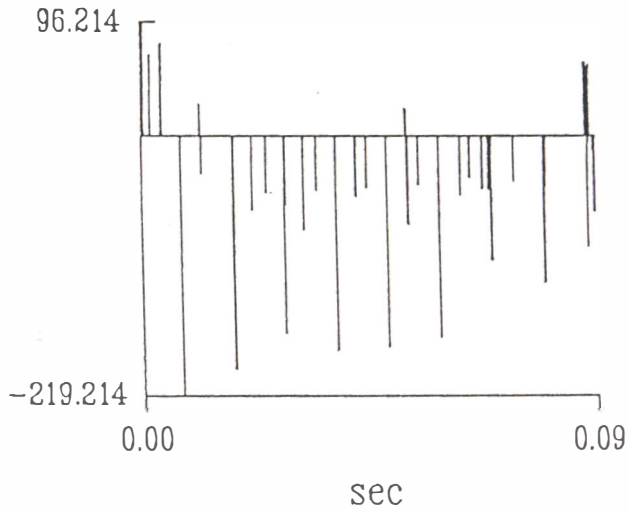


Figure 6.6: *The optimised version of the CLEAN signal of Figure 6.5*

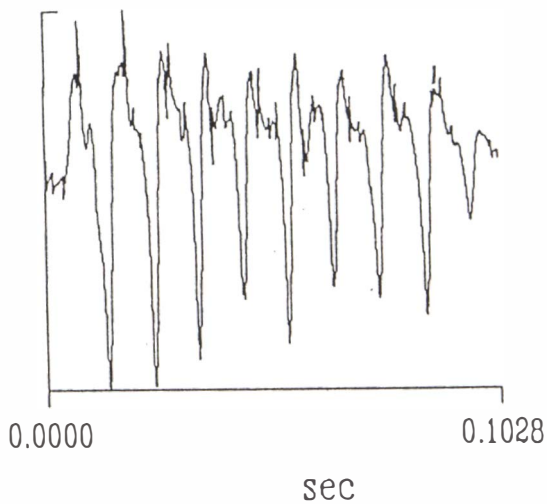


Figure 6.7: *The reconstructed speech by convolving the optimised CLEAN signal (Figure 6.6) with the SAA signal of Figure 6.4.*

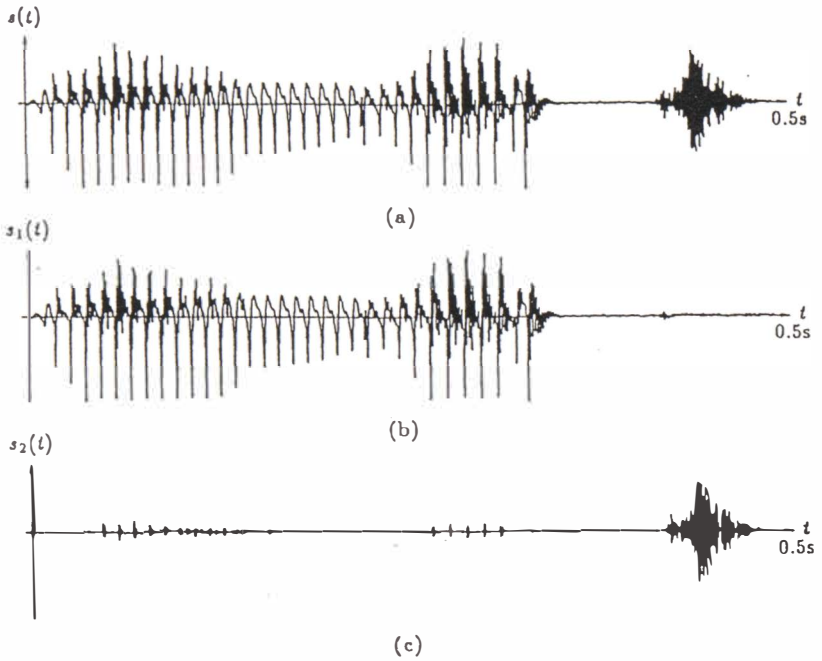


Figure 6.8: **a**: *A section of speech (the word "raindrops"),* **b**: *Low frequency sub-band 0-2.5kHz,* **c**:*High frequency sub-band 2.5-5kHz. Reproduced with permission from (Thorpe, 1990), and with much thanks.*

Since most of the energy of the unvoiced section of speech are concentrated in the high-frequency sub-band, while that of the voiced section are concentrated in the low-frequency sub-band, filtering a speech segment into two sub-bands presents an effective way of separating the voiced and unvoiced sections of speech.

Figure 6.8a shows a section of speech containing both voiced and unvoiced parts. The two sub-bands are shown in Figures 6.8b and c respectively. Notice how the voiced and unvoiced sections shown here are neatly allocated to the two frequency bands, with only a small amount of energy in the low and high frequency sub-bands during the unvoiced and voiced sections of the utterance respectively.

6.1.4.5 A practical speech encoding scheme

Figure 6.9 shows the block diagram of the complete SAA/CLEAN low data rate speech encoding scheme. Referring to Figure 6.9, the speech signal is first separated into two sub-bands. All the following processing is then carried out on each sub-band. The SAA processing proceeds according to the algorithm detailed in §6.1.1. The SAA signals are subsequently edge-extended as described in §6.1.4.2, so that their end-points are of zero amplitudes. CLEAN is then performed in the manner described in §6.1.2 and §6.1.4.3. The resulting CLEAN pulses are subsequently encoded in a way which reflects their statistical structures (namely the probability distribution function and the autocorrelation function of the amplitudes of the CLEAN pulses,

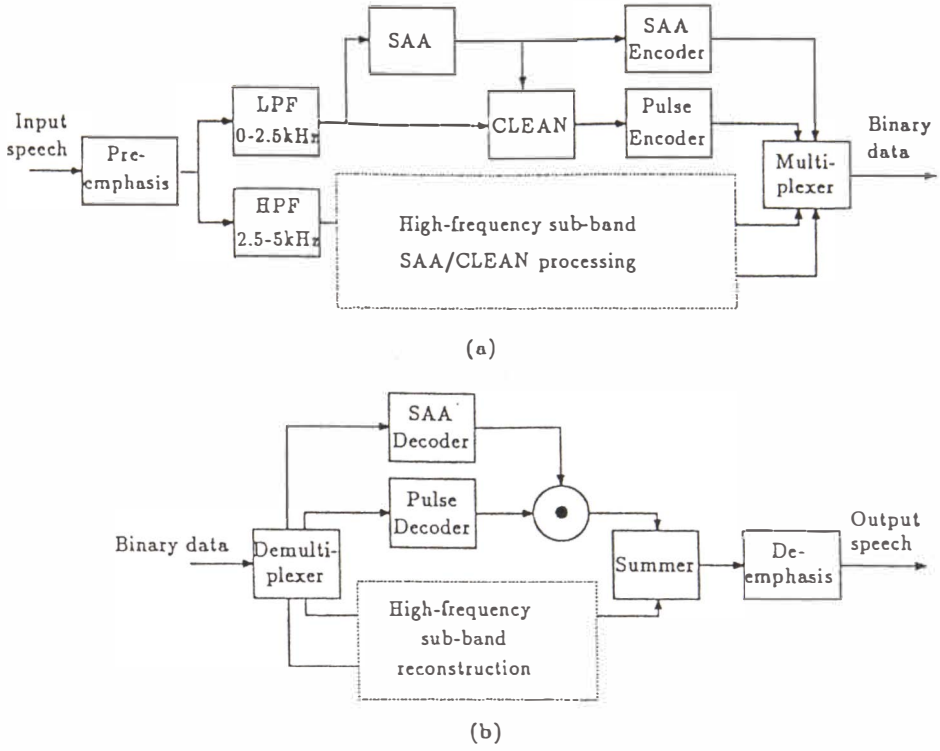


Figure 6.9: *Block diagram of the complete SAA/CLEAN low data rate speech encoding and reconstruction scheme.* **a**: *Speech encoder.* **b**: *Speech decoder.*

and the same two functions of the time interval between each CLEAN pulse), thus ensuring the efficiency of the coding scheme (Thorpe, 1990, Pp 170).

Table 6.1 shows the relationship between the total number of CLEAN pulses occurring in one second (expressed in pulse per second, pps), in each sub-band at various bit rates. The table indicates that, on the average, between 8.2 and 10.2 bits is required to encode each pulse. Notice that, from Table 6.1, the higher band is encoded at a lower bit rate. This is achieved by restricting the number of pulses to a smaller value than the low frequency band, and by encoding the pulse amplitudes with fewer bits.

The SAA signals, which are computed once for each sub-band of an utterance, are stored separately from the CLEAN pulses. The SAA signals require only some 800 bits to code (if the SAA signal is of duration 100 samples and is quantised to 8 bits) which, since it is averaged over the duration of the utterance, adds little to the total data rate.

The reconstructed speech utterance is then formed by, firstly, decoding all the CLEAN pulses and the SAA pulses in each sub-band; secondly, reconstructing the two sub-bands in the manner described in §6.1.3 and finally, adding the reconstructed speech of each sub-band together (see Figure 6.9).

Sentence ID	R_p (pps)		R (bit/s)	Average bits/pulse
	Low	High		
TC7KB	522	253	7133.5	9.2
AC7KB	424	256	6587	9.7
TC11KB	974	378	11082	8.2
AC11KB	789	345	10102	8.9
TC13KB	974	378	13030	9.6
AC12KB	789	345	11530	10.2
TC17KB	1297	626	17300.5	9.0
AC16KB	1124	644	16296	9.2

Table 6.1: *Relationship between rate of occurrence of CLEAN pulses (denoted by R_p, and pps means pulse per second) in each of the two sub-bands and the final coding rate (R) in bits/s required to encode the pulses in both frequency bands. Adapted from Table 5.2 of (Thorpe, 1990).*

6.2 The modified SAA/CLEAN speech coding scheme

The success of the original SAA/CLEAN coding scheme as detailed in §6.1 and Thorpe (1990) depends, to a certain extent, on the assumption that speech can be modelled by the source-filter model (Fant, 1960), *i.e.* the ideal convolution model as described by Equation (6.1), which is repeated here as Equation (6.12), for convenience.

$$s(t) = g(t) \odot v(t) \tag{6.12}$$

In Equation (6.12), $s(t), g(t), v(t)$ represents respectively, the speech signal, the glottal pulse signal and the impulse response of the vocal tract.

However, as in all real world signals such as speech, the ideal convolution model is inadequate. Hence, a more accurate model of speech signal is described by

$$s(t) = g(t) \odot v(t) + c(t) \tag{6.13}$$

where $c(t)$ represents any deviations (from the ideal convolution model) of the real speech signal that is being modelled.

The deviations or contaminations, $c(t)$, can sometimes cause the SAA/CLEAN algorithm to become unstable (see §6.2.1). This instability problem (Cornwell, 1983; Thorpe, 1990) is a well known property of the CLEAN algorithm. The stability of the CLEAN algorithm can be improved by employing a technique originally developed by Cornwell (1983) for image processing applications. I have applied this technique to modify the SAA/CLEAN speech coding scheme described previously (see §6.1). This approach is novel (to speech coding) and hence, constitutes part of my original research.

6.2.1 Instability in CLEAN

The instability of the CLEAN algorithm is best explained in the frequency domain. Thus, taking the Fourier transform of Equation (6.13) gives

$$S(f) = G(f) \times V(f) + C(f) \tag{6.14}$$

where $S(f), G(f), V(f)$ and $C(f)$ are respectively, the spectra of the speech signal $s(t)$, the glottal pulse $g(t)$, the vocal tract response $v(t)$ and the additive contaminations $c(t)$.

In the ideal case where there is negligible contamination *i.e.* $c(t) = 0$, CLEAN essentially performs an inverse filtering operation (Bates and McDonnell, 1989, Pp83), where $\frac{1}{G(f)}$ is the inverse filter. Thus, the spectrum of the vocal tract response can be obtained by

$$V(f) = \frac{S(f)}{G(f)} \tag{6.15}$$

If the contamination $c(t)$ is not negligible, the estimated vocal tract response $\hat{V}(f)$ from the CLEAN algorithm in Fourier domain will have the equation:

$$\hat{V}(f) = \frac{S(f)}{G(f)} - \frac{C(f)}{G(f)} \tag{6.16}$$

which can be re-written as

$$\hat{V}(f) = V(f) - \frac{C(f)}{G(f)} \tag{6.17}$$

Examination of the behaviour of Equation (6.17) at high frequencies yields interesting results. As the spectrum of the contamination $C(f)$ is generally very wide band (*i.e.* non-zero at high frequency), while the spectrum of the $G(f)$ is narrower (compared to that of the contamination $C(f)$), this means that at higher frequencies, $\frac{C(f)}{G(f)}$ tends to swamp $V(f)$ (Bates and McDonnell, 1989, Pp83) because $G(f)$ is almost zero at these frequencies. When this occurs, a CLEAN pulse of large magnitude appears in the CLEAN signal (see Figure 6.10).

6.2.2 Solution to the instability problem

In image processing application, Cornwell (1983) devised a so-called "top-hat" technique to stabilise the CLEAN algorithm (Thorpe, 1990; Bates and McDonnell, 1989). This is effected by adding a constant term to $G(f)$, so that the denominator in $\frac{C(f)}{G(f)}$ is never zero. The term "top-hat" is appropriate since adding a constant term to $G(f)$ is equivalent to adding an impulse or a spike to $g(t)$ in the time domain.

I have modified the SAA/CLEAN coding technique by adding a "top-hat" to the (estimated) glottal pulse. The result of this modification is illustrated in Figure 6.11. Figure 6.11(a) shows a "spike" top-hat is added to the glottal pulse of Figure 6.4 to produce what I call the top-hat glottal pulse. Using the top-hat glottal pulse as the kernel in the CLEAN deconvolution produces the CLEAN signal of Figure 6.11(b). Notice that the CLEAN signal is now less spiky *i.e.* more stable, in particular, the two pulses with unusually large amplitudes have disappeared.

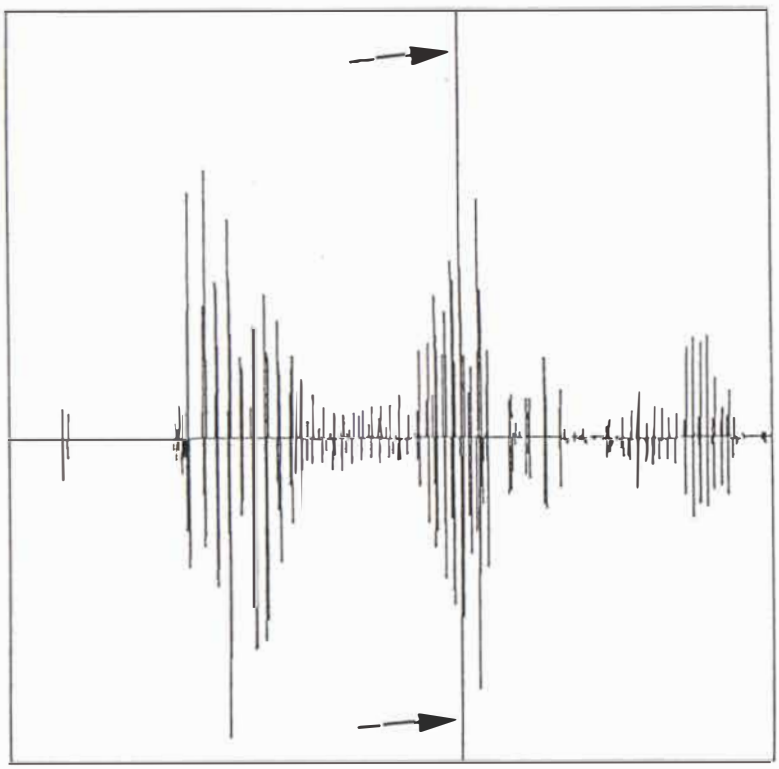


Figure 6.10: *A CLEAN signal illustrating the instability problem in the CLEAN process. The arrows indicate the unusually large-amplitude pulses caused by the instability. These give rise to "spikes" in the reconstructed speech when the CLEAN signal is convolved with the SAA pulse. See Figure 6.12.*

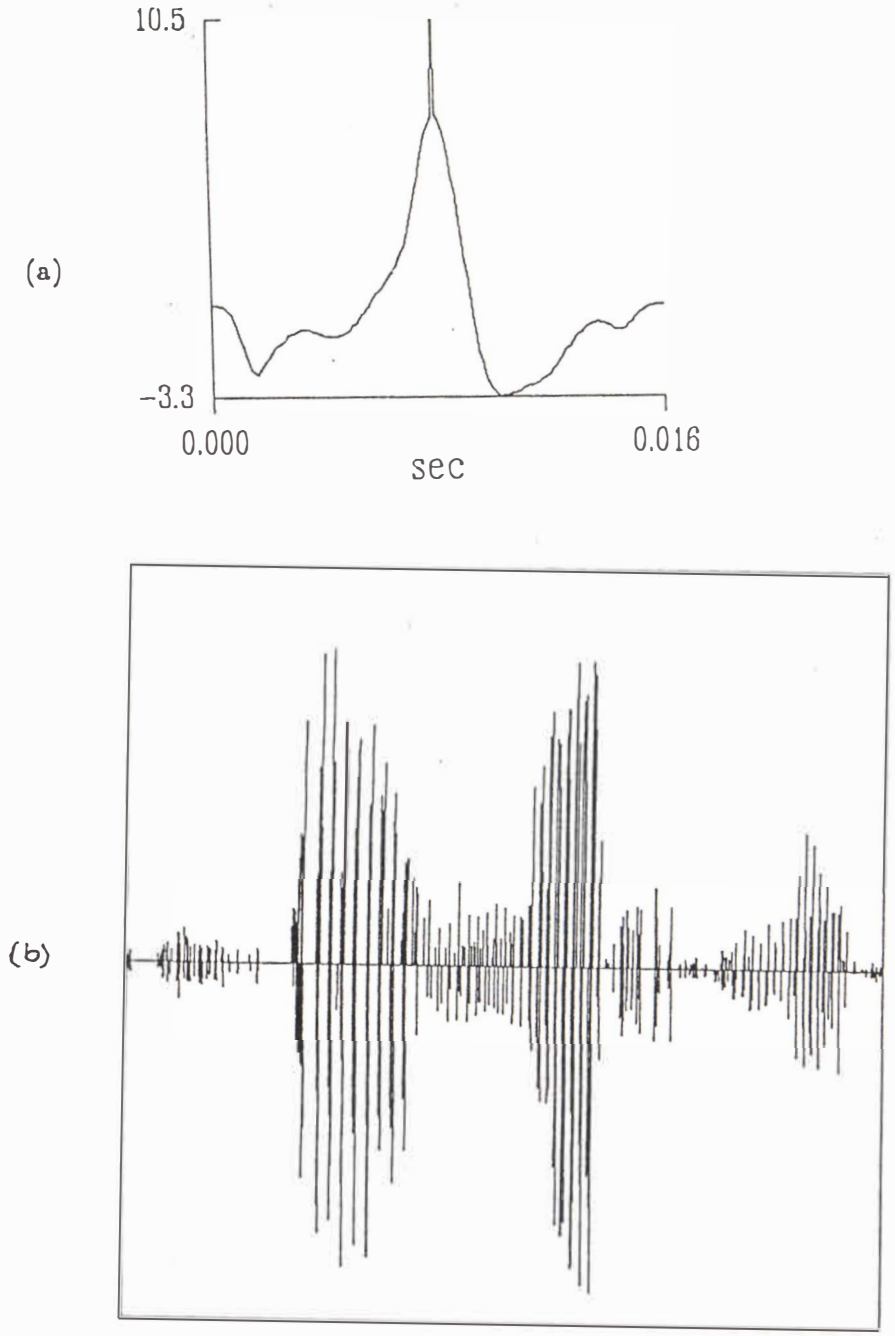


Figure 6.11: *Effects of top-hat operation on CLEAN signal. (a). The "top-hat" glottal pulse, where a spike is added into the glottal pulse. (b) The CLEAN signal produced using the top-hat glottal pulse as the kernel in the CLEAN deconvolution. Notice that the CLEAN signal is now less spiky, in particular, the two pulses with unusually large amplitudes have disappeared.*

The reconstructed speech, using the *original* SAA/CLEAN coding scheme and the *modified* SAA/CLEAN coding scheme, together with the original speech are reproduced here in Figure 6.12 for comparison. It can be observed that the reconstructed speech using the modified SAA/CLEAN scheme resembles more of the original speech. In particular, the extra spikes depicted in Figure 6.12(b) have been removed by using the top-hat glottal pulse in the CLEAN algorithm, as shown in Figure 6.12(c).

6.2.3 System overview

The overall coding system used for the experiment is shown in Figure 6.13.

The input speech is pre-filtered using a low-pass filter with the 3dB cut-off frequency at 4.5KHz and a rolloff of 48 dB/octave. The speech is sampled at the rate of 10 KHz and digitised by a 12 bit A-D converter. It is then divided into two frequency bands to separate out the voiced and unvoiced part of the speech. The cut-off frequency of the low-frequency band (voiced speech) is at 2.5 kHz while the pass-band for the upper frequency band is from 2.5 kHz to 5 kHz (unvoiced speech).

The speech in each of the two subbands is then processed using shift-and-add to extract the glottal pulse, which is then appropriately modified (as described in §6.1.4.2). The kernel of the CLEAN process is then formed by adding a top-hat to the modified glottal pulse, CLEAN is then performed in the manner described in §6.1.2. The CLEAN pulses and the glottal pulse signal are then encoded.

At the decoder end, the CLEAN pulses and SAA signals of the two sub-bands are separately decoded. From the decoded CLEAN pulses and the SAA signals of the two sub-bands, the speech waveform of each sub-band is then reconstructed by a convolution process as described in §6.1.3, and the resultant waveforms are then added to produce the reconstructed speech. After passing through a low-pass filter having a 3 dB cut-off frequency of 4.5 KHz with a roll off of 48 dB/octave, the reconstructed speech is then converted to its analogue counterpart with a 12 bit D/A converter.

6.2.4 The low frequency band

The input speech is passed through the 256 taps low-pass Finite Impulse Response (FIR) filter with a 3dB cut-off frequency of 2.5 kHz. The frequency response of the low-pass FIR filter is shown in Figure 6.14.

The speech is pre-emphasised using first order differentiation. The SAA algorithm is then performed on the input speech to extract an approximation to the glottal pulse of the speaker. In the SAA algorithm, the size of the window where a peak is searched is 128 samples. This value is chosen because it is slightly over the typical pitch period at a sampling frequency of 10 KHz.

The SAA signal is then edge-extended using a 4 term Blackman-Harris window (Harris, 1978) of 32 samples long. The window is first divided into two halves, and each half is used for the extensions of the starting edge and the trailing edge of the SAA signal respectively (see Figure 6.4). Following the edge-extension of the

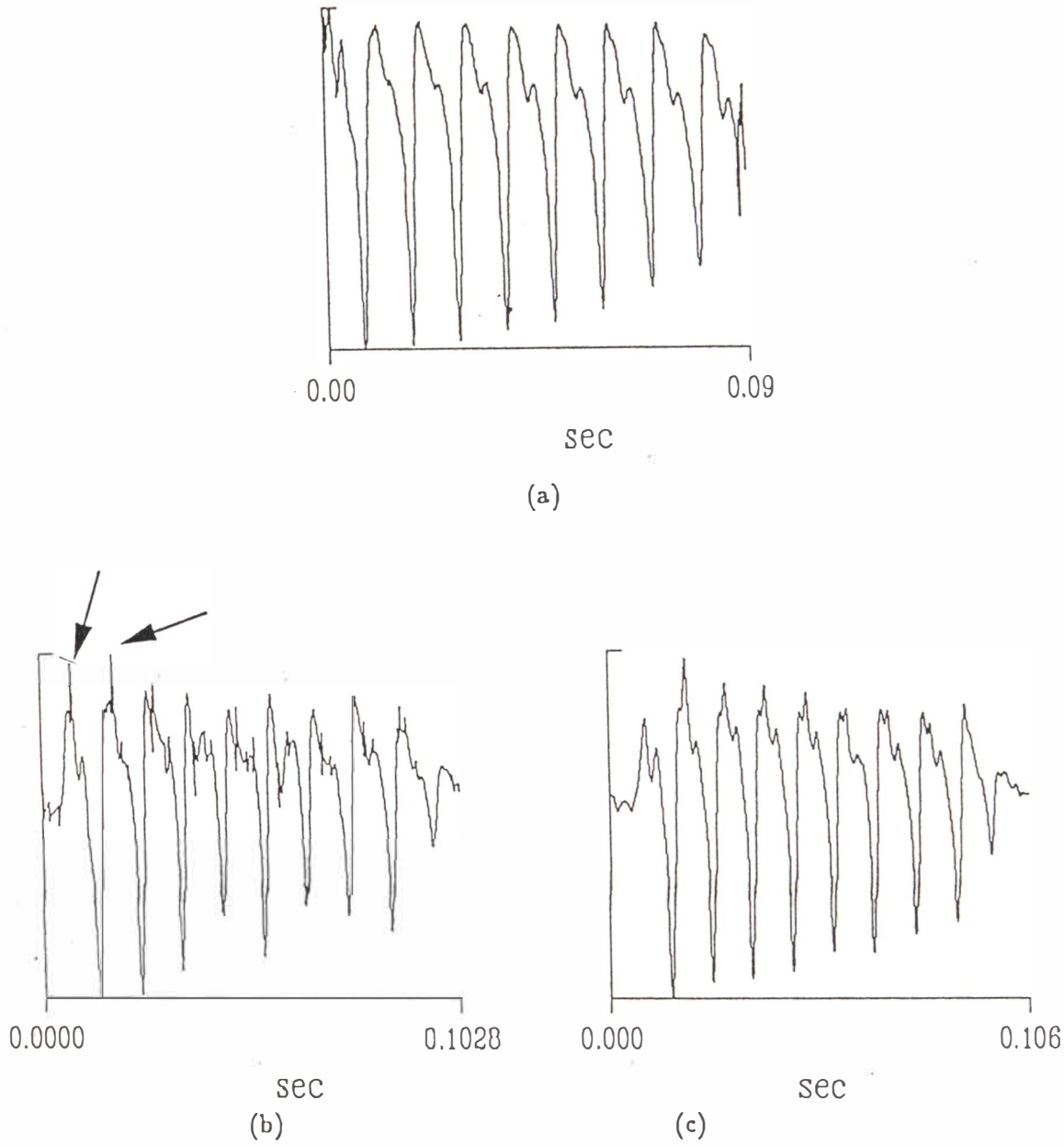
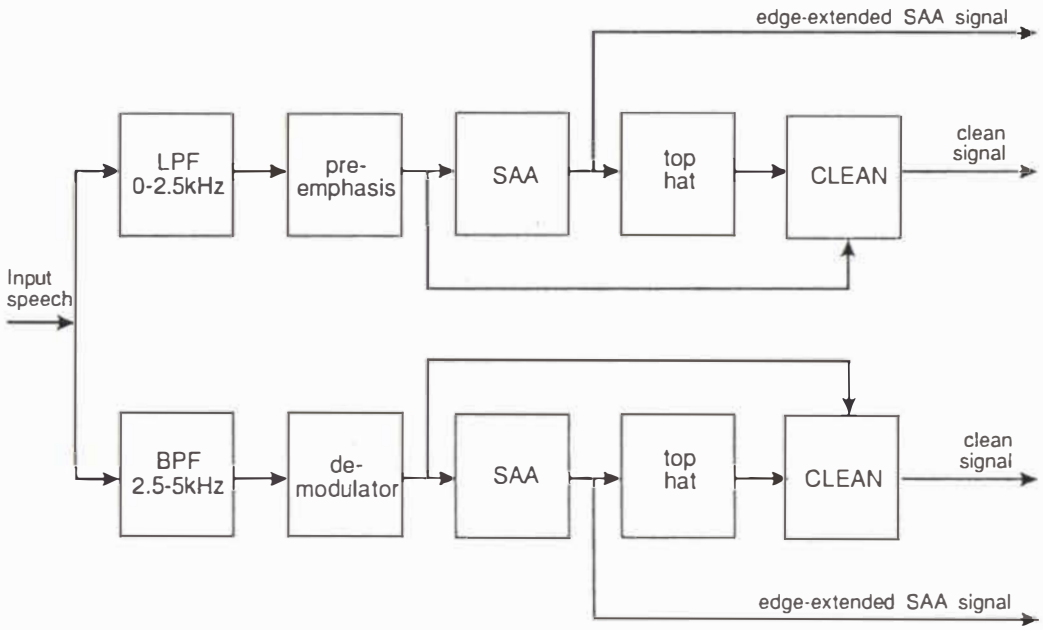


Figure 6.12: *Effects of top-hat operation on the reconstructed speech. (a) The original speech. (b) The reconstructed speech without top-hat. (c) The reconstructed speech using the modified SAA/CLEAN coding scheme with edge-extended and top-hat glottal pulse as the kernel in the CLEAN deconvolution. Notice that the modified scheme successfully removes the spikes (which is not present in the original signal) in the first two peaks of the speech signal.*

- SAA/CLEAN ENCODER:



- SAA/CLEAN DECODER:

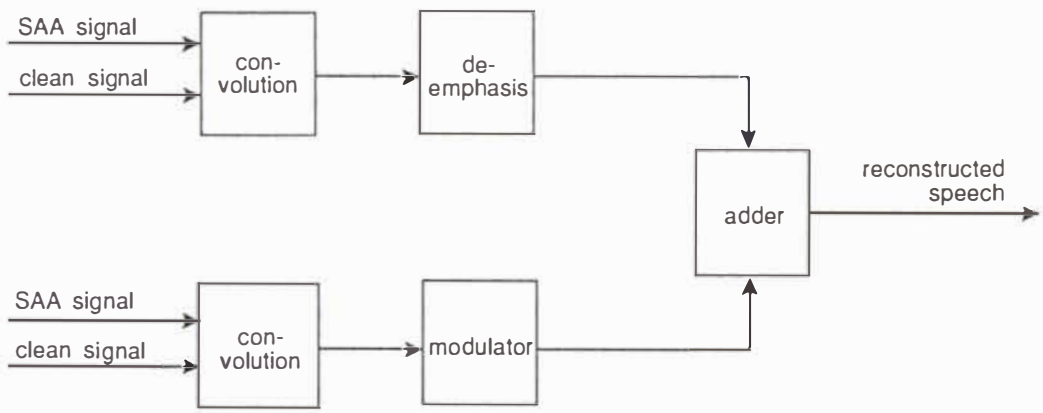


Figure 6.13: *Block diagram of the CLEAN speech coding system with top-hat.*

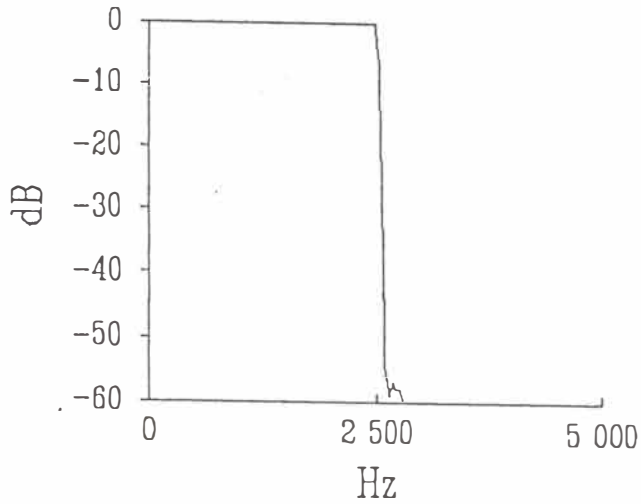


Figure 6.14: *Frequency response of the 256 taps low-pass FIR filter.*

SAA signal and the addition of a top-hat to the edge-extend SAA signal, the CLEAN algorithm is then used to obtain the CLEAN signal. The SAA signal and the CLEAN signal of the low-frequency band are shown in Figure 6.15.

The sparse CLEAN signal and the edge-extended SAA signal are encoded. By controlling the number of pulses, N_p (defined in §6.1.2), in the CLEAN signal, the coding rate of the system can be changed (§6.2.6).

At the decoder end, the SAA signal is then convolved with the CLEAN signal. The output waveform is then de-emphasised using first order integration.

6.2.5 The high frequency band

In the high-frequency band of the system, the input speech is first filtered using a 256 taps band-pass FIR filter with a bandwidth of 2.5 kHz centred at 3.75 KHz (or a pass band from 2.5 KHz to 5 kHz). The frequency response of the band-pass FIR filter is shown in Figure 6.16.

The speech is then translated to the baseband (0 - 2.5 kHz) by a demodulation process. The baseband signal is subsequently low-pass filtered by the 256-taps FIR filter whose frequency response has been shown in Figure 6.14.

The CLEAN signal for this subband is obtained in the same way as the low-frequency band. The SAA signal and the CLEAN signal for the high frequency band are shown in Figure 6.17.

The edge-extended SAA signal and the CLEAN signal are then encoded. At the

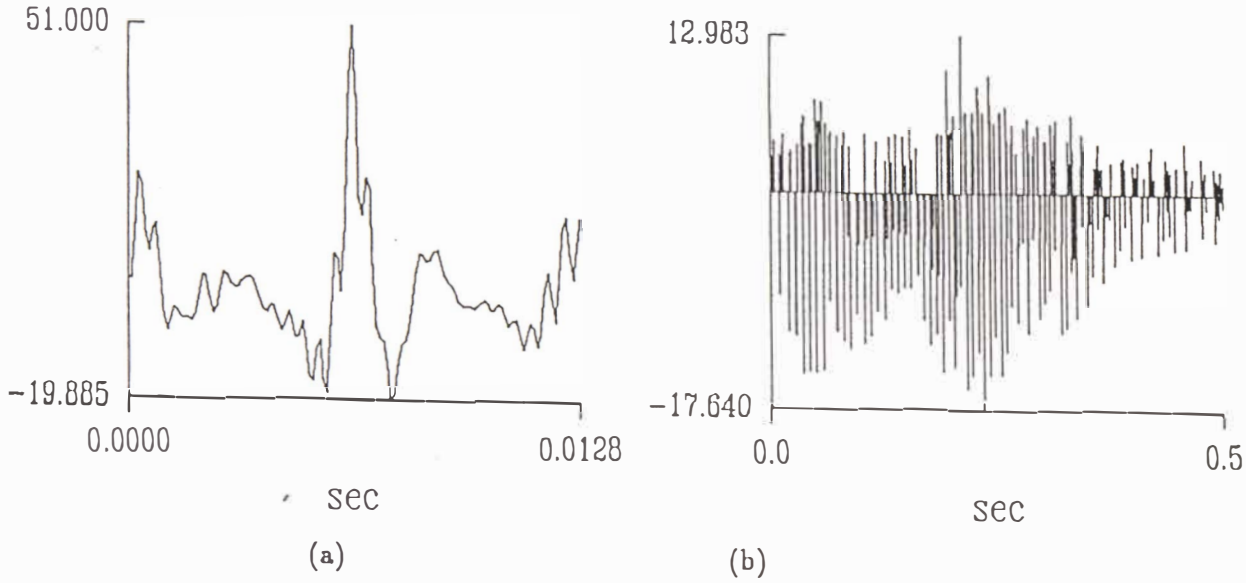


Figure 6.15: *(a) The SAA signal of the low-frequency band and (b) the CLEAN signal of the low-frequency band.*

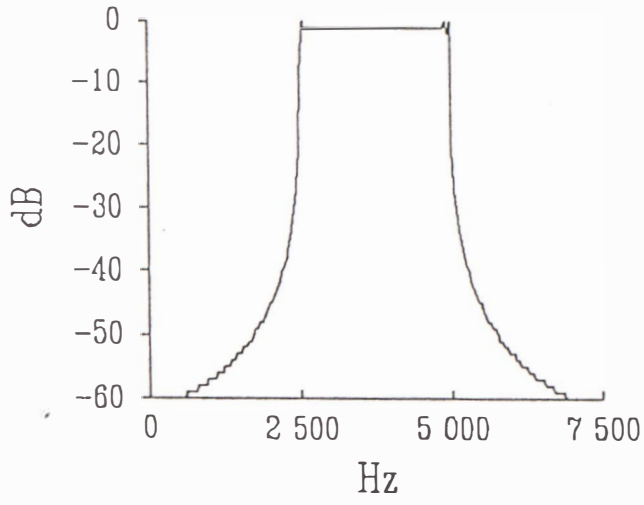


Figure 6.16: *The frequency response of the 256 taps band-pass FIR filter.*

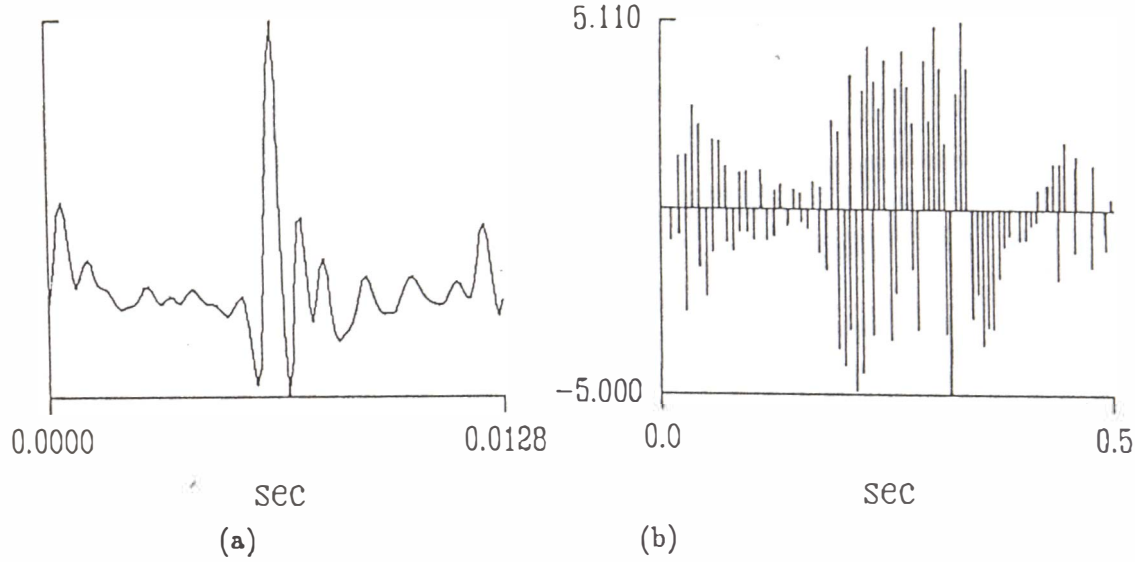


Figure 6.17: *(a) The SAA signal of the high frequency band and (b) the CLEAN signal of the high frequency band.*

decoder, the decoded SAA and the CLEAN signals are re-convolved. The output waveform of this convolution is then translated back up in frequency to 2.5 to 5.0 kHz. The output waveform is then filtered with the same 256 taps band-pass FIR filter whose frequency response has been shown in Figure 6.16.

Finally, the reconstructed version of the original input speech waveform is obtained by adding the output waveforms of both frequency bands (see Figure 6.13).

6.2.6 Controlling the bit rate

In the (modified) SAA/CLEAN algorithm, the bit rate is controlled by various parameters described below. The default settings of these parameters are summarised in Table 6.2.

The bit rate is primarily controlled by limiting the number of pulses, N_p, in each segment (see §6.1.2). The higher this parameter is set, the more pulses results from the CLEAN process. This in turn means that more pulses have to be encoded, and thus the bit rate is increased. To achieve a maximum bit rate of 16 kbits/s (kbps), the maximum allowable number of pulses, N_p, in a segment are 10 and 5 for the low-frequency and the high-frequency bands respectively. In the case of 8 (4) kbps encoding rate, the corresponding values for N_p are 5 (3) and 2 (1) respectively.

The bit rate is also controlled to a lesser extent by the setting of the threshold *thres* (see §6.1.2). The threshold *thres* is a value below which, if and when the

Bit Rate (kbps)	N_p		bits/pulse
	Low	High	
16	10	5	12
8	5	3	12
4	3	1	12

Table 6.2: *Default values of various parameters of the Modified SAA/CLEAN speech coding scheme at various coding bit rates. N_p refers to the maximum allowable number of pulses in each segment of speech being coded. The default values for the segment length and segment spacing have been set at 150 samples and 100 samples respectively at all coding bit rates. The default value of the loop gain factor, α, is set at 0.6.*

maximum amplitude of the residual signal, $r_i(n)$ at the i^{th} iteration falls, causes the CLEAN process to stop "identifying" more CLEAN pulses (§6.1.2). Hence, a low value for *thres* indicates that more CLEAN pulses are identified, which implies a higher bit rate. In all the experiments reported herein, the threshold *thres* is set at 0.2.

Another parameter which affects the number of CLEAN pulses is the loop gain factor, α. A small value of α implies that, for each iteration of the CLEAN process, only a small proportion of the glottal pulse *i.e.* the kernel is *subtracted* from the original signal. This results in more iterations and more (non-zero) CLEAN pulses. The loop gain factor has been set at 0.6 in all my experiments.

Each sample of the CLEAN pulses have been coded with 12 bits, with 8 bits being allocated to encode the amplitudes of the pulses and 4 bits to encode the intervals between pulses. This represents a conservative allocation of bits/pulse compared to the original SAA/CLEAN speech coding scheme (see Table 6.1). However, this is compensated by linearly encoding the amplitudes and the intervals between pulses. The SAA signals of both frequency bands have been allocated 8 bits/sample.

6.3 Evaluation procedure

To evaluate the quality of the reconstructed speech, four American and three New Zealand utterances have been used (see Table 6.3 and Table 6.4). The utterances have been coded into 16 kbits, 8 kbits and 4 kbits per second. Three different shapes of top-hat, *i.e.* spike or impulse, rectangular and triangular have been tested.

The utterances from the Americans have been extracted from the DARPA-TIMIT database (Price *et al.*, 1988) while the utterances from the New Zealanders have been recorded in an anechoic chamber using an AIWA CM-53 microphone and amplified by an AIWA F990 amplifier. The 3dB frequency responses of the microphone and the tapedeck (when employing Dolby C noise reduction, and optimised bias) have been set at 50 Hz and 13 KHz and at 20 Hz and 18 kHz respectively (Tan, 1990; Brieseman *et al.*, 1987).

In order to prevent aliasing effect (Haykin, 1989), each utterance has been filtered by a low-pass filter with a cutoff frequency of 4.5 kHz and a rolloff of 48 dB/octave.

Utterance No	Speaker ID	Sex	Accent	Phrase
1	NZ-A	M	New Zealand	AWAY
2	USA-A	M	USA	RAG
3	USA-A	M	USA	SUIT
4	USA-A	M	USA	RAG
5	USA-B	F	USA	WARDROBE
6	NZ-B	M	New Zealand	AWAY
7	NZ-C	M	New Zealand	FUN

Table 6.3: *List of utterances used to study the effects of top-hat coding. The phrases are identified in Table 6.4.*

Label	Phrase
AWAY	We were away a year ago.
RAG	Don't ask me to carry an oily rag like that.
SUIT	She had your dark suit in greasy wash water all year.
WARDROBE	Her wardrobe consists of only skirts and blouses.
FUN	It's been fun working with you this morning.

Table 6.4: *The phrases corresponding to the utterances listed in Table 6.3.*

The utterances have been digitised using a 12 bit analogue-to-digital (A/D) converter at a rate of 10 000 samples per second. The digitised utterances are then processed using the modified SAA/CLEAN algorithm (described earlier).

After processing, the reconstructed utterances are converted into their analogue counterparts using another personal computer based signal processing package (called SIGPLAY). The utterances are then recorded on a TDK C-60 audio cassette using a National RX-C47F cassette recorder via an AW 19 amplifier.

6.3.1 Reference signals

In order to quantify the "equivalent perceptual distortions" (Kitawaki and Nagabuchi, 1988) introduced by the Modified SAA/CLEAN speech coding algorithm, a reference signal has been used as a bench-mark in the evaluation of the quality of the reconstructed speech. The reference signal is constructed by adding controlled amount of noise to the original speech.

Two types of noise, additive noise signal and multiplicative noise signal, have been considered. These two noise signals are fundamentally different with regard to generation and perceptual effect (IEEE, 1969). The additive random noise signal has not been used because the quality of the reconstructed speech using the Modified SAA/CLEAN algorithm is very different from that produced by the additive, uncorrelated, broad spectrum noise (Tan, 1990). As a result, the subjects find it difficult to make consistent comparisons and their responses may vary widely owing to different and fluctuating subjective criteria (Schroeder, 1968b).

This difficulty is overcome by using the multiplicative noise signal (described below), which, when added to the original speech signal, introduces a distortion which is perceptually similar to the reconstructed speech of the Modified SAA/CLEAN coding system (Tan, 1990).

The multiplicative noise reference signal is described by

$$\hat{s}(t) = s(t) + k.n(t).s(t) \tag{6.18}$$

where $s(t)$ is the original speech, $n(t)$ is a white, uncorrelated (to the speech signal $s(t)$) noise (Schroeder, 1968b) and k is a gain factor which is expressed as a percentage of the maximum value of the input speech. Notice that in Equation (6.18), the original signal, $s(t)$, is first *multiplied* by the noise signal, $n(t)$, to produce what is termed the *multiplicative* noise. The reference signal, $\hat{s}(t)$, is then obtained by adding a scaled (by the gain factor k) version of the multiplicative noise to the original signal, $s(t)$.

6.3.2 Subjective assessment

Assessment of the *quality* of encoded speech is a difficult and long standing problem (Jayant, 1990) because the quality of the (encoded) speech is, ultimately a human perception and is therefore subjective. While most distortion or distance measures (refer to §3.2.1.2 for specific examples of these) do indeed correlate well with subjective assessment of speech quality, especially at a coding rate of 32 kbps and above,

they (the distortion measures) are not suitable at lower (that is 16 kbps and below) bit rate because the correlation is poor (see §3.2.1.2). Since one of the object of this thesis is to study the performance of the SAA/CLEAN speech coding at 16 kbps and below, I have decided to use the frequently used mean opinion score (MOS) (IEEE, 1969; Jayant, 1990) to assess the quality of the encoded speech.

Two sessions of subjective assessments of the recorded reconstructed utterances have been conducted in a language laboratory. The laboratory is equipped with a master cassette player located in front and about 40 sets of stereo ear-phones fitted to rows of cubicles separated by perspex glass. The volumes of the ear-phones are adjustable.

Twenty three and twenty six human subjects attended the first session and the second session respectively.

The IEEE [1] recommended practice for speech quality measurement has been closely followed. Specifically, the category-judgement method using Mean Opinion Score (MOS) (IEEE, 1969) is used. Each session consists of two phases; familiarisation and evaluation.

6.3.2.1 Familiarisation

During the familiarisation phase, the listeners are first presented with a questionnaire as shown in Figure 6.18.

The listeners are then presented with the range of qualities of utterances to be encountered.

Eight sets of reconstructed sentences are evaluated in each session. The listeners are advised that the original speech is always played first at the beginning of each set of experiment, and that it is given a MOS of 5, *i.e.*, of perfect quality. This serves as some sort of "reference" or "anchor" points for the listeners.

6.3.2.2 Evaluation

In the evaluation phase, the listeners are requested to grade the quality of the speech according to the five (1 being unacceptable quality and 5 being excellent quality) categories listed at the top of the questionnaire.

6.3.3 Experimental parameters

Each set of experiments consists of either varying the height of the top-hat (in the case of the "spike" top-hat) or varying the width of the top-hat (in the case of the triangular top-hat and the rectangular top-hat where the heights are fixed at 60% of the height of the SAA pulse). These parameters (*i.e.* the height or the width of the top-hat are increased in steps of equal amounts.

The utterances are coded into 16 kbits, 8 kbits and 4 kbits per second using the top-hat method. The height of the added spike is varied from 0 to 120 percent of the maximum value of the glottal pulse at these coding bit rates.

[1] The Institute of Electrical and Electronics Engineers, Inc., U.S.A

SUBJECTIVE ASSESSMENT

PERSONAL DETAILS (for statistics purposes only)

sex : ☐ age : ☐

INSTRUCTIONS

You will be played some sentences of recorded speech.
Please judge the quality of the speech according to the score ranging from 1 to 5 as described below :

5 = excellent; perfect quality
4 = good, telephone (toll) quality
3 = fair, distortions present but not obvious
2 = poor, but still intelligible
1 = unsatisfactory, voice cannot always be identified

Please be consistent with your judgements.
First sentence of every experiment is rated with the score of 5 as shown :

EXPT 1	1	2	3	4	5	6	7	8
score :	5							

EXPT 2	1	2	3	4	5	6	7	8
score :	5							

EXPT 3	1	2	3	4	5	6	7	8
score :	5							

EXPT 4	1	2	3	4	5	6	7	8
score :	5							

EXPT 5	1	2	3	4	5	6	7	8
score :	5							

EXPT 6	1	2	3	4	5	6	7	8
score :	5							

EXPT 7	1	2	3	4	5	6	7	8
score :	5							

EXPT 8	1	2	3	4	5	6	7	8

Figure 6.18: *Questionnaire used for the subjective assessment.*

Rectangular and triangular top-hats are also experimented at a coding rate of 8 kbits per second. The heights of these two top-hats are set at 60 per cent of the height of the SAA glottal pulse while the width of the base is varied. In the case of the rectangular top-hat, the width of the rectangular pulse is varied from 0 to 12 percent with respect to the width of the original glottal pulse. For the triangular top-hat, the base of the triangle is varied from 0 to 12 per cent relative to the width of the glottal pulse.

6.4 Experimental results

Twenty three and twenty six human subjects attended the first and second session of the subjective assessments respectively. Nineteen (19) males and four (4) females attended the first session while seventeen (17) males and Nine (9) females attended the second session. The human subjects consisted of New Zealanders and Asians. The results of the subjective assessments are then averaged to give the mean opinion score (MOS) (IEEE, 1969). These results are discussed in §6.4.1 - §6.4.4.

6.4.1 The multiplicative reference signal

The speech-amplitude-correlated-noise (see Equation (6.18)) is introduced to the original speech to produce a reference signal for the subjective assessments. Figure 6.19 shows the average MOS of the two sessions for the reference signal versus the percentage distortion, k, given in Equation (6.18). Thus the quality of the CLEAN speech coding can be expressed in "opinion equivalent distortion", which is the distortion subjectively equivalent to a given level of the multiplicative noise. The result also shows that the greater the amount of added noise, the lower the mean opinion score. Thus, the result is intuitively appealing.

6.4.2 Effects of the spike top-hat

The MOS results at the coding rate of 16 kbits, 8 kbits and 4 kbits per second using the "spike" top-hat of different heights are shown in Figure 6.20 and Figure 6.21.

Figure 6.20 shows the MOS for the utterances of the New Zealanders. From Figure 6.20, it can be observed that the addition of the spike top-hat did not affect the quality of the reconstructed speech significantly. In fact, it shows that the reconstructed speech without the top-hat has the highest MOS. At the coding rate of 16 kbits per second, a MOS of 3.33 and "opinion equivalent distortion" of less than 0.02 percent are obtained in the case of reconstructed signal without top-hat. While in the cases of the coding rates of 8 kbits and 4 kbits per second, MOS of 2.83 and 2.09 are obtained respectively. The "opinion equivalent distortion" at these two coding rates are 0.015 and 0.050 percent respectively.

Figure 6.21 shows the MOS for reconstructed speech from the Americans, using spike top-hat. Again, the results show that the addition of a spike top hat has very little effect on the MOS. Furthermore, the MOS at the three different coding rate

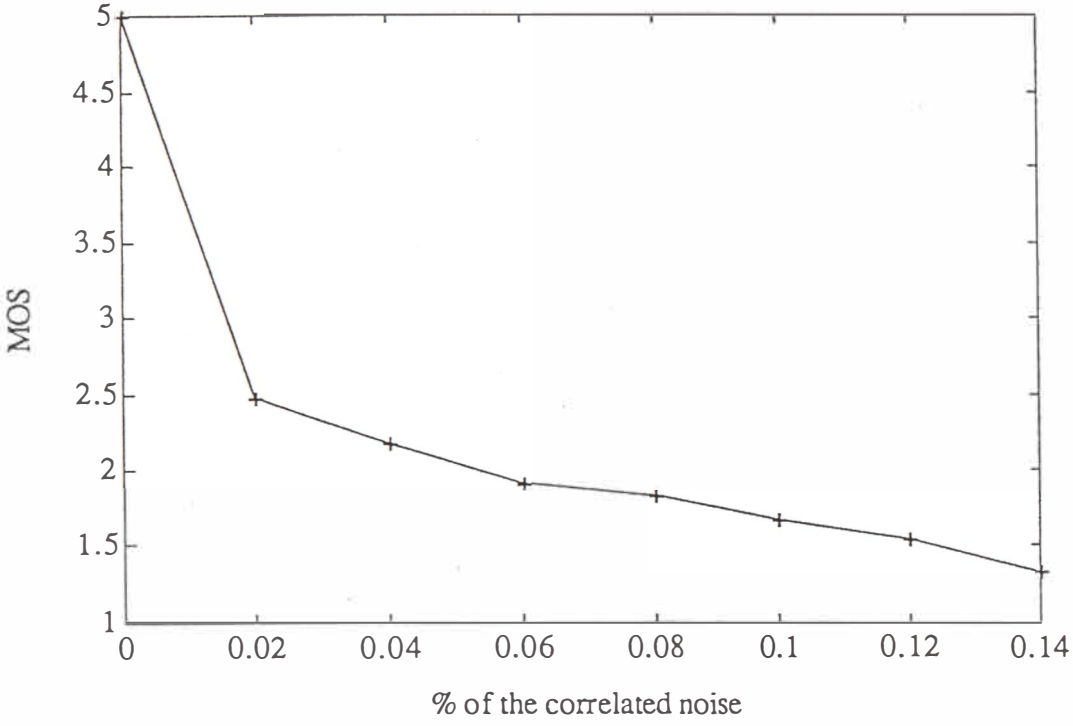


Figure 6.19: *The MOS of the speech-amplitude-correlated-noise reference signal.*

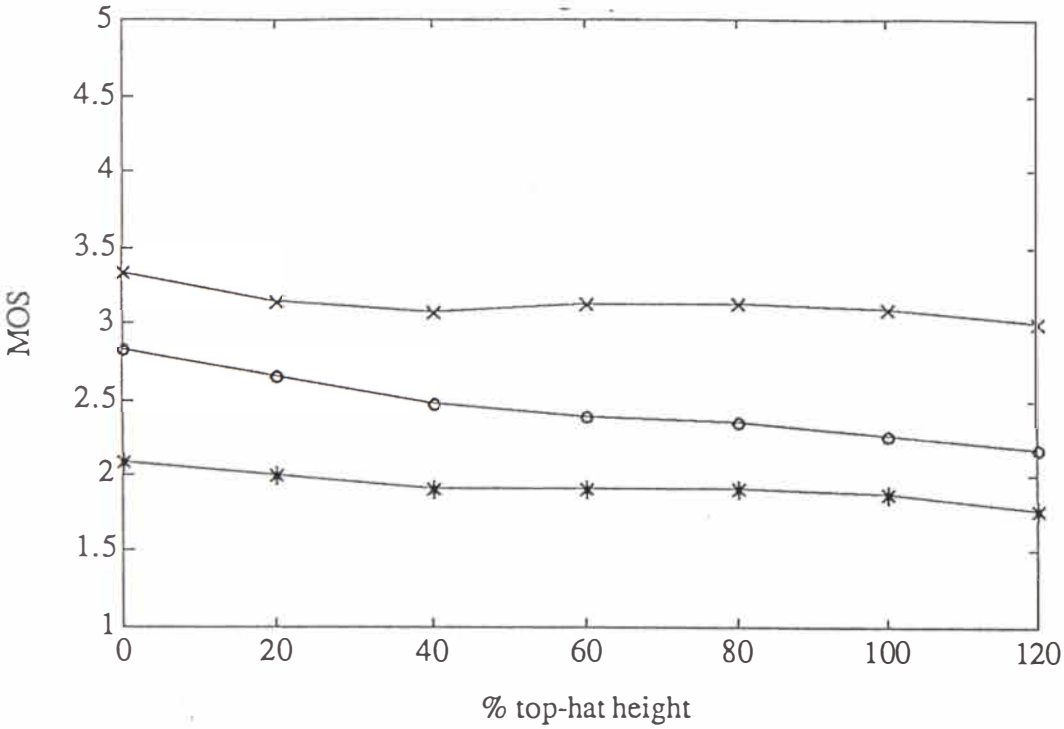


Figure 6.20: *Mean Opinion Score (MOS) for reconstructed speech from a New Zealander, at a coding rate of 16 kbits (×), 8 kbits (o) and 4 kbits (∗) per second using 'spike' top-hat.*

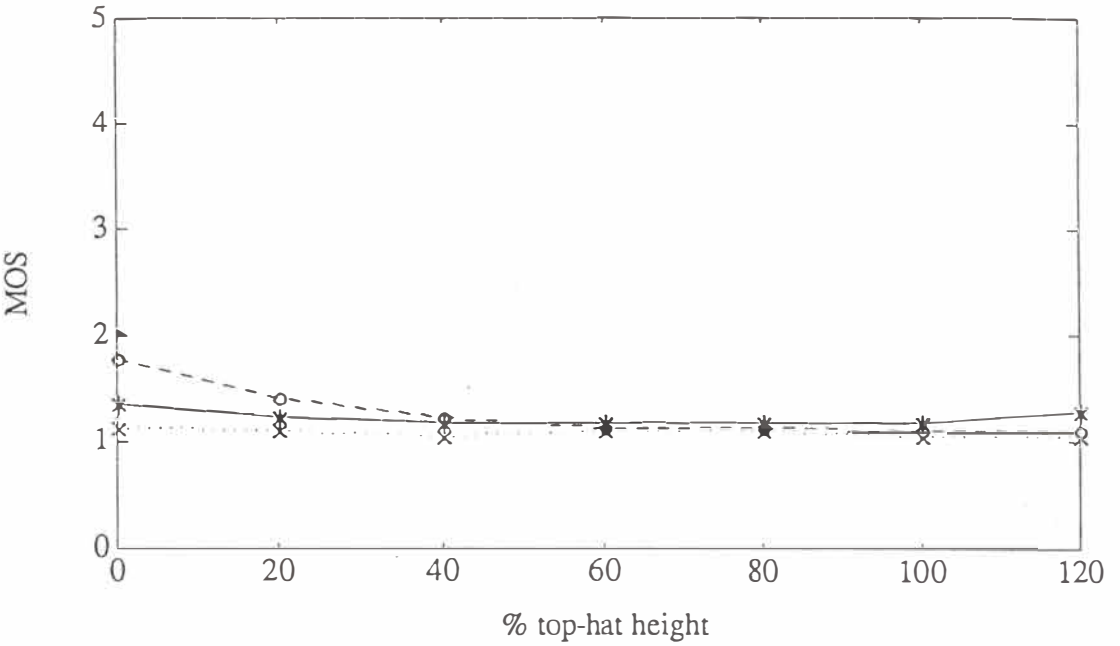


Figure 6.21: *Mean Opinion Score (MOS) for reconstructed speech from the Americans, at a coding rate of 16 kbits (×), 8 kbits (○) and 4 kbits (✳) per second using 'spike' top-hat.*

are very close together, as compared to the New Zealand speech, where there is good separation between the MOS at these coding rate. See Figure 6.20 and Figure 6.21.

Notice that the New Zealand utterances have a higher rating of MOS at all the three coding rates tested. This is due to the higher quality of the New Zealand utterances when compared to that of the American utterances.

6.4.3 Effects of the rectangular top-hat

The MOS results of the reconstructed speech using the rectangular top-hat, with a height which is 60 per cent of the maximum height of the SAA signal, are shown in Figure 6.22. The speech is coded at 8 kbits per second. Comparing the MOS results for the rectangular top-hat and with those for the spike top-hat of Figure 6.20, it can be seen that the MOS of both categories is almost the same, with the maximum difference of 0.5 in MOS scale between the two graphs. The MOS for the utterances from New Zealanders are again higher than those of Americans.

It is also interesting to note that, as the width of the rectangular "top-hat" increases, the MOS of the American shows an upward trend *i.e* the quality is perceived to be better. Surprisingly, this trend is reverse in the case of the New Zealander. Thus, this may provide a means whereby a machine can be used to identify the accent of a speaker. This, if proves successful, may provide a useful communication tool for those without hearing ability.

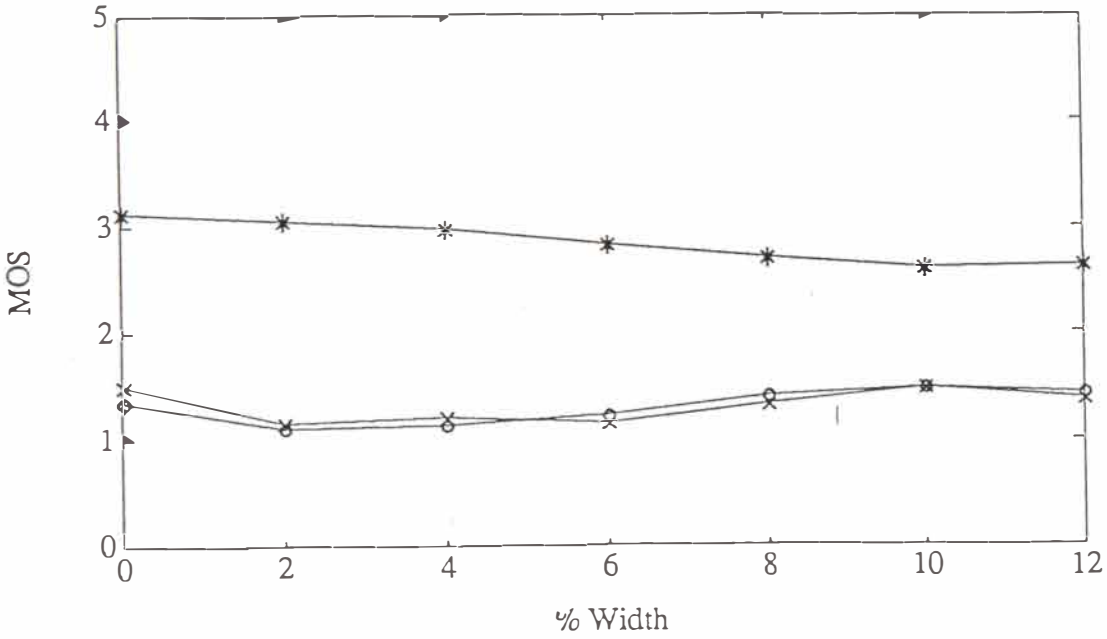


Figure 6.22: *Mean Opinion Score (MOS) for reconstructed speech from two Americans (×, ○) and one New Zealander (✱), at 8 kbps using rectangular top-hat.*

6.4.4 Effects of the triangular top-hat

The MOS results for the quality of the reconstructed speech with the addition of the triangular top-hat are shown in Figure 6.23. The triangular top-hat has a height of 60 percent of the maximum value of the glottal pulse. The width of its base is varied from 0 to 12 percent of the width of the glottal pulse.

Figure 6.23 shows that the MOS results for the three different shapes of top-hat, it is clear that the shapes of the top-hat did not affect the quality of the reconstructed speech significantly.

6.5 Summary

The SAA/CLEAN speech coding scheme has been investigated, modified and subjectively evaluated. It is found that while the SAA/CLEAN coding algorithm produces "good" quality speech at 16 kbps, the encoded speech contains unpleasant "click" sounds caused by the instability of the SAA/CLEAN coding algorithm.

The SAA/CLEAN coding scheme is modified by adding a "top-hat" to the kernel of the coding scheme. This is found to successfully remove the unpleasant "click" sounds. It is also found that the time waveform of the decoded speech (using the modified scheme) exhibits a better resemblance to the original time waveform of the speech.

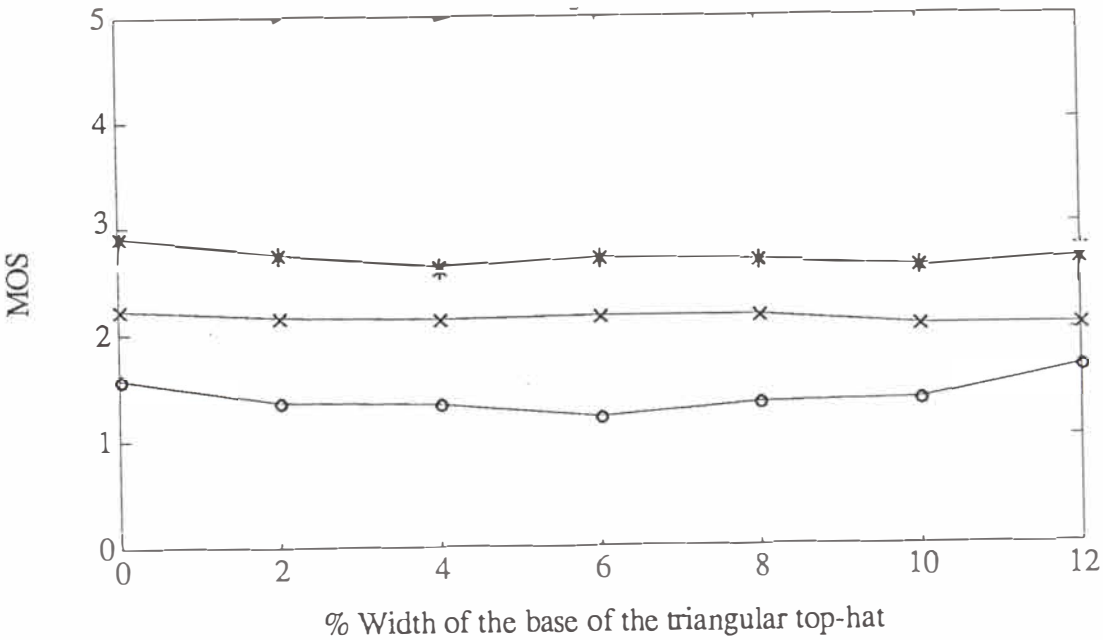


Figure 6.23: *Mean Opinion Score (MOS) for reconstructed speech from two New Zealanders (* $*$, $\times$ *) and an American (o) at 8 kbps using triangular top-hat.*

Four American phrases and three New Zealand phrases have been used in the subjective evaluation of the quality of the reconstructed speech using the modified scheme. Various heights of the added top-hat spike have been used. The effects of the addition of the rectangular and triangular top-hat have also been evaluated.

The evaluation has been carried out by 49 human subjects in a language laboratory and the Mean Opinion Score (MOS) (IEEE, 1969) is calculated. It is found that the MOS decreases as the coding bit rates decreases. At 16 kbits per second, a MOS of 3.33 is obtained. It is also found that the shape of top-hat does not change the quality of the speech significantly.

In order to characterise the distortion (of the original speech waveform) introduced by the (modified) SAA/CLEAN coding scheme, two types of noise (*i.e. additive* and *multiplicative*) have been added to the original speech. Informal listening tests strongly indicates that the distortion introduced by the (modified) SAA/CLEAN coding scheme is multiplicative rather than additive. This finding is surprising and may have implications in further ways of improving the (modified) SAA/CLEAN coding scheme.

In addition, by comparing the MOS scores of the reconstructed speech with the original speech contaminated with known amount of multiplicative noise, the distortion introduced by the Modified SAA/CLEAN coding system can be quantified. For example, in the case of 8 kbps coding rate with "spike" top-hat, a MOS of 2.83 is achieved. This value of MOS is subjectively equivalent to adding a 0.015 percent

(of the height of the glottal pulse) of speech-correlated-noise to the original speech.

Chapter 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH AND DEVELOPMENT

"The trouble with you people, is you try to write a whole thesis in one sentence."
(R. H. T. Bates, Professor of Electrical & Electronic Engineering)

This thesis discusses various aspects related to human speech in Chapter 1. The thesis also reviews and provides, from Chapter 2 to Chapter 4, the background for some of the established techniques in speech processing in general, and speech coding as well as speech recognition in particular. Each of the chapters from Chapter 1 to Chapter 6 concludes with a summary recounting the main issues discussed in the chapter.

In this final and concluding chapter of the thesis, a summary of the findings of the research on the recognition by computer and the coding of speech signal, and a commentary on the main features of each of the recognition and coding techniques studied is provided in §7.1. In addition, several promising avenues for future research and development are outlined in §7.2.

7.1 Conclusions

7.1.1 Speech Recognition

One of the areas studied as part of this Ph.D. research is speech recognition by computer. In the study, I have treated the problem of speech recognition as a pattern

recognition one rather than a neural-psychological one. Furthermore, the much more difficult problem of "speech understanding" (whatever this means) is not dealt with.

Specific conclusions and the findings of the experiments that I have conducted have already been reported in Chapters 5 and 6 and are not repeated here. The following paragraphs relate to more general "expert opinion" which results from the materials that the author has consumed (literally? I mean read) and the author's experimental experience.

From the study, it is concluded that speech recognition by computer is practicable in structured conversation involving a fixed set of vocabulary, but still a long way from being able to recognise free-flowing daily conversation between human beings. This is because during our free-flowing daily conversation, one tend to fill in the "gaps" and apply the context in which the conversation is held. This is a task which human beings seems to be able to do naturally and without much effort but a computer is incapable at the moment.

The implication of the last paragraph is that we are still a long way off from being able to converse with a computer the same way human beings do a lot among ourselves. It also means that a speaker who intends to use an automatic speech recogniser must be cooperative and be prepared to enunciate each word carefully. It is very difficult if not near impossible to train a computer to do the same. However, the current technology can be developed into useful tools. The avenues for further research and development are discussed in §7.2.

I have used the digits zero to nine in all my recognition experiments. This set of vocabulary, despite its relatively small size, contains some characteristics which makes it a particularly difficult and challenging recognition problem. For example, all but one (*i.e.* the digit "seven") of the words in the set are of one syllable long. There are also pairs of words which contain parts which sounded very similar. For instance, the words "one" and "nine" both contain the same phoneme at the end. This, perhaps, explains the fact that these two words are often confused by the computer speech recogniser, *i.e.* "one" is often (mis)-identified as "nine" and vice-versa.

Three different techniques for automatic speech recognition have been discussed to varying depths. These are the hidden Markov model, the neural network and the dynamic time warping techniques (see Chapter 4). The input to these three recognition system is generally a sequence of multi-dimensional features which is extracted from the waveform of the word being recognised while the output is one of the word in the vocabulary set of the recogniser.

It is worthwhile pointing out that these recognition techniques accomplish the task of recognising the unknown word through different approaches. For example, the dynamic time warping technique (§4.3.1) through non-linearly stretching the time axis of an *input* (as defined in the previous paragraph) against a set of reference features, thereby reducing the adverse effect that different speed of speaking may have on the recognition accuracy; the HMM technique (§4.3.2) through capturing the statistical structure of the word being recognised and choose the most probable candidate; while the neural network technique (§4.3.3) through modelling the way the human brain achieves cognition.

7.1.2 Speech coding

As intimated in §2.2.6, §2.2.7 and Chapter 6, the SAA/CLEAN technique is an innovative scheme for speech coding. Its main attraction over other speech coding technique such as those employing linear predictive coefficients is that there is no need for accurate pitch estimation. This means that one potential source of error (arising from inaccuracy in pitch estimation) is eliminated.

The SAA/CLEAN coding scheme (§2.2.6, §2.2.7 and §6.1) has been evaluated using subjective and objective means (Thorpe, 1990). These tests have shown that the technique is viable and that it gives comparable results to those published in the literature (Jayant, 1990; Kitawaki and Nagabuchi, 1988; Kitawaki *et al.*, 1984). However, it is found that the reconstructed speech using the SAA/CLEAN coding scheme contains unpleasant "click" sounds caused by the instability of the modified SAA/CLEAN coding scheme (§6.2). These "click" sounds are manifested as the unusually large peaks in the CLEAN signal depicted in Figure 6.10. By modifying the SAA/CLEAN coding technique in the way described in §6.2, it is found that the large peaks have been removed along with the "unpleasant" click sounds.

It should be reiterated at this point that the assessment of the qualit of encoded speech is a difficult problem. This is because the quality of the encoded speech is ultimately a human perception and therefore subjective. From the literature (Jayant, 1990; Kitawaki and Nagabuchi, 1988; Kitawaki *et al.*, 1984) and the experience of the author and the speech group at the Department of Electrical & Electronic Engineering, University of Canterbury, it has been found that objective distortion or distance measures (such as those described in §3.2.1.2) gives a good correlation between the quantitative distortion measures and the subjective mean opinion score, especially at 16 kbps and above (Jayant, 1990; Thorpe, 1990). However, at lower bit rate, it is concluded that the most reliable and dependable measure available at present remains the MOS test as outlined in §6.3.2.

7.2 Suggestions for future research and development

Having completed the research presented in this thesis, it is now obvious that there are many ways in which it can be improved and expanded upon. These are outlined in §7.2.1 and §7.1.2.

7.2.1 Speech recognition

While the accuracy of the speech recognition system as implemented by the author gives comparable results to that of most published results (Grant, 1991), its main limitation is speed. This can be overcome by implementing the present system in a digital signal processing chip such as the TMS320C30. Our experience (Watson *et al.*, 1988; Bates *et al.*, 1988) indicates that this should enable the speech recognition system to respond within a reasonable time.

As stated at the beginning of this chapter, automatic speech recognition by computer is practical and particularly useful in conversation involving structured or

set formats such as those carried out at an airline ticket booking office. In these conversations, the vocabularies that are most likely used are the months of the year, the time of the day, and place name. Hence, useful systems can be developed to automate the booking of airline tickets, railway tickets and so on.

In addition, future research and development should be geared towards developing tools or aids for people with disability. Colour reader for the blind, voice input unit which can translate voice into commands which activate some hydraulic manipulators or machineries for people with limited physical strength are examples of tools which are within the capabilities of the present technology.

Finally, in the longer term (centuries may be?), automatic translation from one language to another may be made possible by combining speech recognition technique and speech coding technique. Other future possible application such as electronic voice mail and dictaphones for the typing personnel would enhance the quality of human life in general.

7.2.2 Speech coding

As explained by Shannon (1948), efficient coding philosophy requires that an event which is more likely to occur be encoded with fewer bits than one which is less likely to occur [1]. This means that one should exploit the probability distribution function (or equivalently the histogram) of the CLEAN signal (refer to Figure 6.10) so that the CLEAN pulses which occurs more frequently are coded with less bits. However, this increases the complexity of the SAA/CLEAN coding scheme and consequently the amount of time required to encode a segment of speech can become unacceptably long.

The last paragraph suggests two possible future research directions depending on whether the requirement is for efficiency (as defined in the footnote) or speed. If the requirement is for efficiency of the modified SAA/CLEAN coding scheme, then one should examine the histogram of the amplitudes of the CLEAN pulses and allocate more bits to those pulses which occur less frequently. However, if the requirement is for faster (*i.e.* less delay) coding scheme but without undue compromise in the quality of the encoded speech, then the research should be directed towards implementing the modified SAA/CLEAN coding scheme as described in §6.2 in a digital signal processing chip such as the TMS320C30.

Another possible avenue for future research is to incorporate the modulo-PCM coding technique (Ramamoorthy, 1985) into the Modified SAA/CLEAN algorithm. This involves dividing the CLEAN pulses, according to their amplitudes, into 2^N groups and then encoding each group using Pulse Code Modulation technique (see §2.4.1.1).

A further promising avenue for future research is to incorporate vector quantization technique (§3.2) into the Modified SAA/CLEAN technique in order to reduce the coding bit rate even further. This involves the following two steps: firstly, ev-

[1] An efficient code is one which uses the least *average* number of bits to encode a fixed number of symbols in such a way that each symbol is *uniquely identifiable* by a *binary* code. An example of such a code is the Huffman (1952) code.

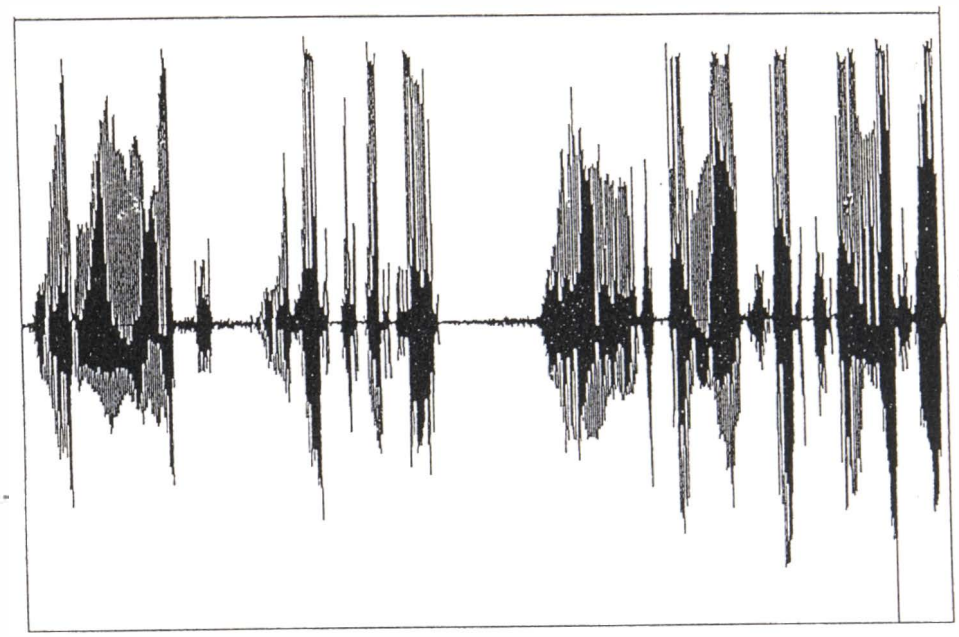


Figure 7.1: *The time waveform of a male speaker uttering the sentence* "When sunlight strikes raindrop in the air, it acts like a prism, and forms a rainbow."

ery ten individual CLEAN pulses are concatenated to form a 10-dimensional vectors and secondly, these 10-dimensional vectors are encoded using vector quantization technique.

The advantage of incorporating the vector quantization technique is now exposed. The present system encodes the *amplitude* of each pulses individually using eight (8) binary digits (or bits). However, if every ten of these pulses are concatenated to form a 10-dimensional vectors and then vector-quantized into 256 ($=2^8$) levels, then only 8 bits is required for every 10 pulses, which represents a saving of an order. This means that instead of needing a total of 80,000 bits to encode 10,000 CLEAN pulses (which is a typical number for all the experiments that the author has conducted), only 8,000 bits is required.

Preliminary informal listening tests indicate that incorporation of the vector quantization technique shows an improvement in quality while maintaining the bit rate of the reconstructed speech at a reasonably low level. Several figures are now depicted to illustrate the effectiveness of this scheme. Figure 7.1 shows the time waveform of a male speaker uttering the sentence "*When sunlight strikes raindrop in the air, it acts like a prism, and forms a rainbow*".

The time waveform as shown in Figure 7.1 is then divided into two frequency bands. SAA/CLEAN operation is subsequently carried out independently on each band. This results in two sets of CLEAN signal, one for each frequency band. Each set of the CLEAN signal is then separated into two buffers: the *amplitude buffer*

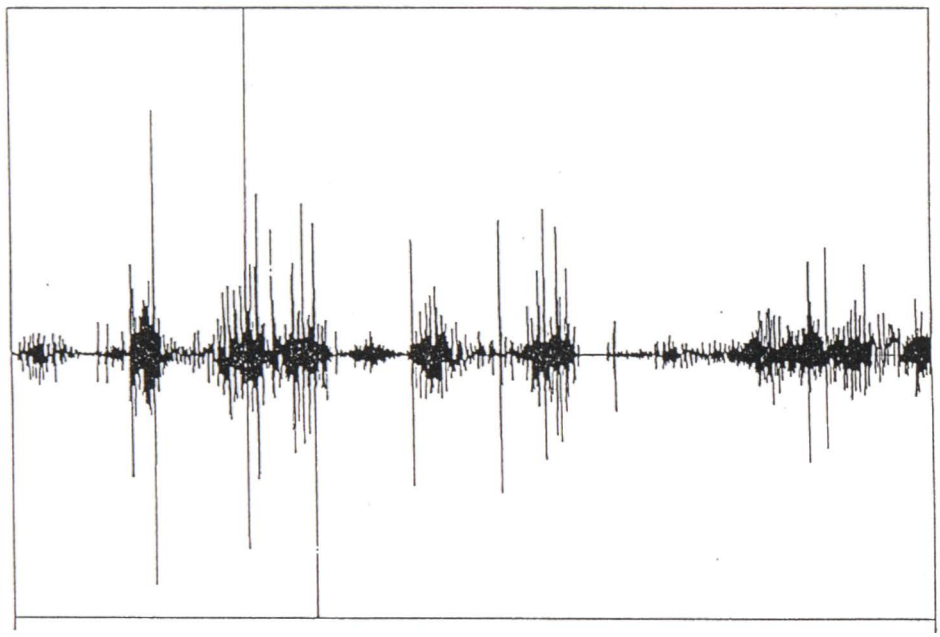


Figure 7.2: *The amplitude of the low frequency band of the CLEAN signal,* AMP1.

which records the height of each CLEAN pulse, and the *time buffer* which records when each pulse occur. The amplitude of the low frequency band of the CLEAN signal, denoted by the symbol **AMP1**, is shown in Figure 7.2.

The signal **AMP1** is then vector-quantized into 256 levels. The resulting signal, denoted by the symbol **256VQAMP1**, is illustrated in Figure 7.3.

The *arithmetic difference* between the signal **AMP1** and **256VQAMP1** is shown in Figure 7.4, which shows the inevitable *quantization error* (§3.2). Notice that Figure 7.2, Figure 7.3 and Figure 7.4 are drawn on the same vertical scale for easy comparison.

Without going into further details, it suffices to say that the same processing is carried out with the *time buffer* of the low frequency band, denoted by **TIM1**, and the *amplitude* and *time* buffers of the *high frequency band* of the CLEAN signal, denoted by **AMP2** and **TIM2** respectively. After some careful manipulation, all the relevant parts of the original waveform (*i.e.* Figure 7.1) are put back together to give Figure 7.5, which is the reconstructed speech waveform. Notice that the original speech waveform (Figure 7.1) and the reconstructed speech waveform (Figure 7.5) are almost indistinguishable. This indicates the effectiveness of the scheme. The effectiveness of the scheme is also further confirmed by informal listening tests which shows that there is no significant difference between the perceived quality of the original speech and the reconstructed speech.

Lastly, I think it is quite safe to say that, in assessing the quality of an utterance, a listener (whether trained or untrained) is almost certainly influenced by a lot of

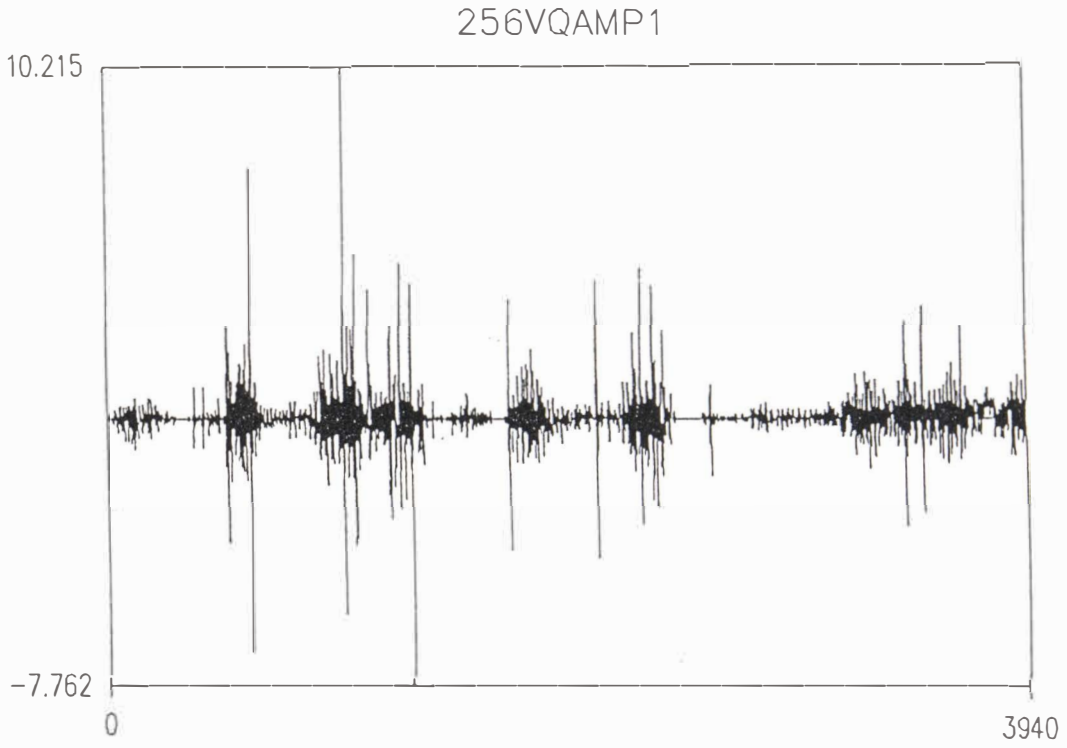


Figure 7.3: *The result of vector-quantizing* **AMP1** *as shown in Figure 7.2 into 256 levels.*

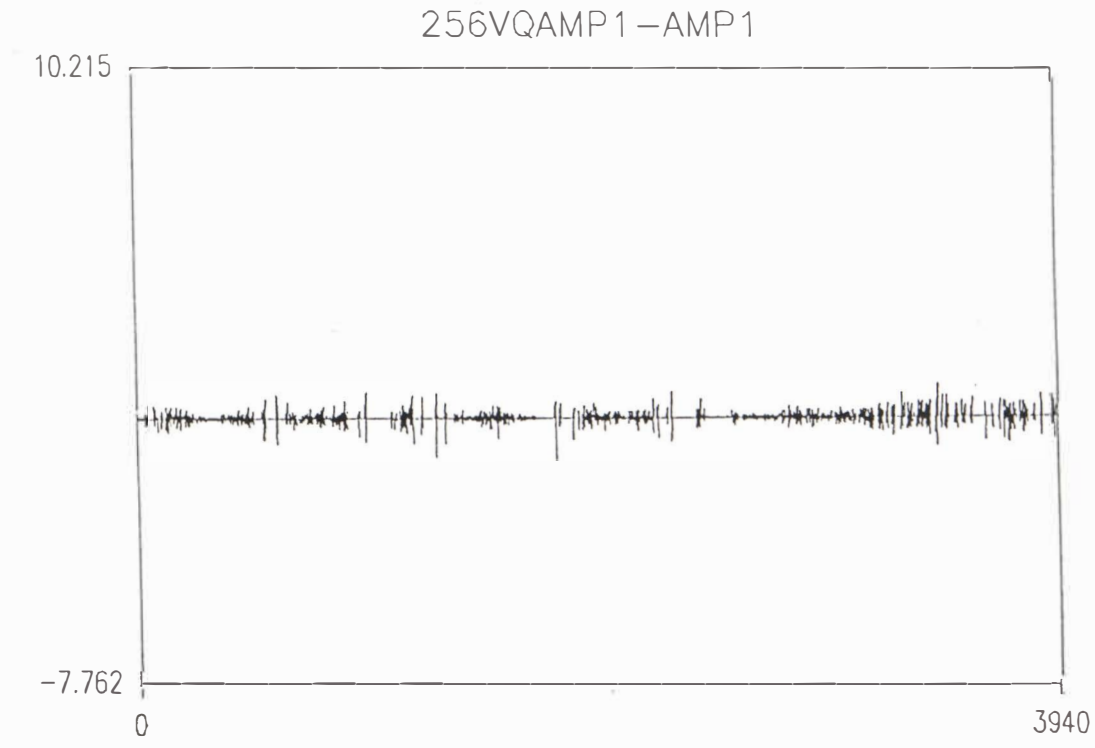


Figure 7.4: *The* arithmetic difference *between* **AMP1** *and its vector-quantized version* **256VQAMP1.**

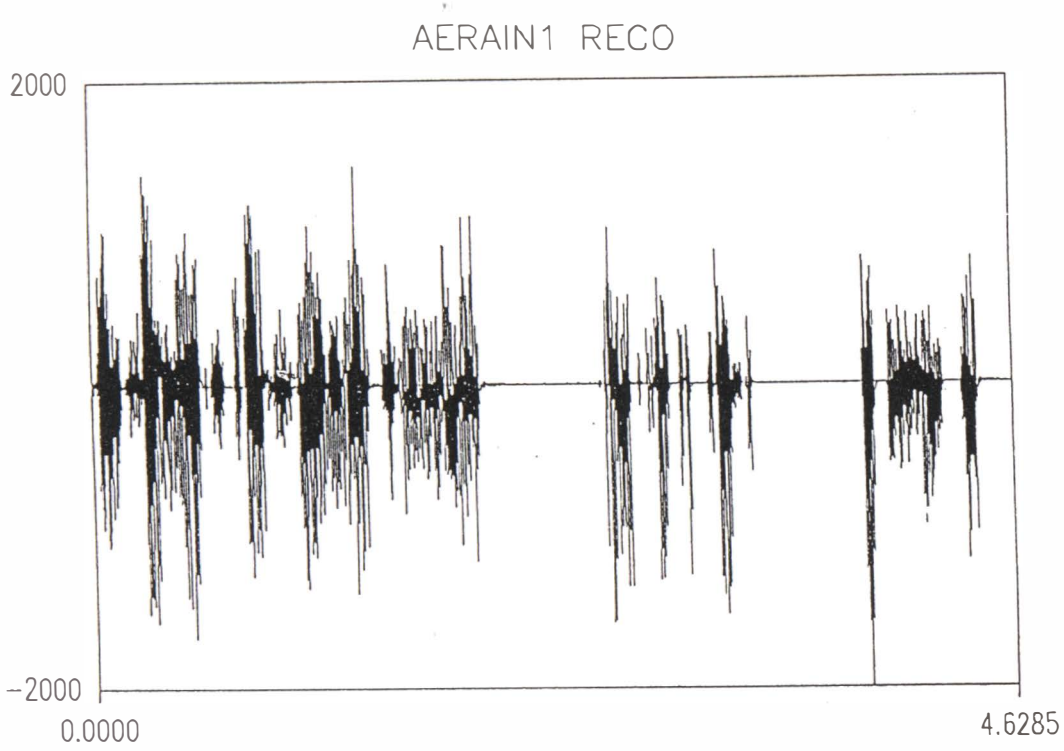


Figure 7.5: *The reconstructed speech waveform.*

factors. For example, if a listener is familiar with the utterance or the voice of the speaker, it is more likely that the listener will give that particular utterance a higher rating, since the listener has a better chance of understanding (or making up, or comprehending) what has actually been uttered. Other factors which may influence the assessment are the educational (whatever this means) background of the listener and the speaker, their cooperativeness, their mood, or even perhaps whether they had any breakfast or not in the morning. It is also quite safe to say that, while a lot is now known about how human perceive the quality of speech (or indeed of anything), the search for an "agreement" on what constitutes "quality" and *the ultimate*(if it exists) objective measure of the quality of speech, is likely to involve the battles of various "experts" from various diverse disciplines such as neuroscientists, psychologists, speech scientists and cognition scientists or any other professions(?) with multi-syllable words. For how long? I, for one, really do not know!

****** THE END ******

References

AINSWORTH, W.A. (1970), 'Estimation of speech synthesiser parameters from acoustic waveforms', *Int. J. Man-Machine Studies*, Vol. 2, Pp. 291–302.

AINSWORTH, W.A. (1988), *Speech Recognition by Machine*, IEE Computing series 12, Peter Peregrinus Ltd. London.

ALIPHAS, A. and FELDMAN, J.A. (1987), 'The versality of digital signal processing chips', *IEEE Spectrum*, June, Pp. 40–45.

ANDREWS, H.L. (1984), 'Speech processing', *Computer*, October, Pp. 315–324.

ATAL, B.S. (1974), 'Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification', *Journal of the Acoustical Society of America*, Vol. 55, No. 6, June, Pp. 1304–1312.

ATAL, B.S. (1985), 'Linear predictive coding of speech', In FALLSIDE, F. and WOODS, W.A. (Eds.), *Computer speech processing*, Prentice Hall, Englewood Cliffs, NJ.

ATAL, B.S. and HANAUER, S.L. (1971), 'Speech analysis and synthesis by linear prediction', *Journal of the Acoustical Society of America*, Vol. 50, No. 2 (part 2), Pp. 637–655.

ATAL, B.S. and REMDE, J.R. (1980), 'A new model of lpc excitation for producing natural-sounding speech at low bit rates', In *Proc. Int. Conf. Acoust., Speech and Signal Processing*, IEEE-ASSP Society, Paris, France, Pp. 614–617.

ATAL, B.S. and SCHROEDER, M.R. (1967), 'Predictive coding of speech signals', *Proc. 1967 IEEE Conf. on Communication and Processing*, Pp. 360–361.

AUSTIN, S.C. and FALLSIDE, F. (1988), 'Frame compression in hidden markov model', In *International conference on speech and signal processing*, IEEE, ASSP Society, New York city, April, Pp. 477–480.

BAHL, L., BROWN, P.F., P. V. deSOUZA, R.L.M. and PICHENY, M.A. (1988), 'Acoustic markov models used in the Tangora speech recognition system', In *International conference on speech and signal processing*, IEEE, ASSP Society, New Yprk, New York, April, Pp. 497–500.

BATES, J.H.T. (1981a), *Applications of modelling and image processing in medicine*, PhD thesis, University of Otago, Dunedin, New Zealand.

BATES, J.H.T. (1981b), *Applications of modelling and image processing in medicine*, PhD thesis, University of Otago, Dunedin, N.Z.

BATES, R.H.T. (1982), 'Astronomical speckle imaging', *Physics Reports*, Vol. 90, Pp. 203–297.

BATES, R.H.T. and MCDONNELL, M.J. (1989), *Image Restoration and Reconstruction*, The Oxford Engineering Science Series, Oxford University Press, paperback ed.

BATES, R.H.T. and ROBINSON, B.S. (1981), 'Ultrasonic transmission speckle imaging', *Ultrasonic Imaging*, Vol. 3, No. 4, October, Pp. 378–394.

BATES, R.H.T., BRIESEMAN, N.P., CLARK, T.M., ELDER, A.G., FRIGHT, W.R., GARDEN, K.L., KENNEDY, W.K., SQUIRES, P.L., THORPE, C.W., TURNER, S.G. and JELINEK, H.J. (1987), 'Interactive speech-defect diagnostic/therapeutic /prosthetic aid', In LETELLIER, J.P. (Ed.), *Real Time Signal Processing X*, Proceedings of SPIE - The International Society for Optical Engineering, 20-21 August, Pp. 131–139.

BATES, R.H.T., CLARK, T.M., ELDER, A.G., GARDEN, K.L., KENNEDY, W.K., SQUIRES, P.L., THORPE, C.W. and WATSON, C.I. (1988), *Speech processing at Canterbury*, Technical Report, Electrical and Electronic Engineering Department, University of Canterbury, Christchurch, New Zealand, March 8.

BAUM, L.E. and PETRIE, T. (1966), 'Statistical inference for probabilistic functions of finite state markov chains', *The Annals of Mathematical Statistics*, Vol. 37, Pp. 1554–1563.

BAUM, L.E., PETRIE, T., SOULES, G. and WEBB, N. (1970), 'A maximization technique occurence in the statistical analysis of probabilistic functions of Markov chains', *The Annals of Mathematical Stastics*, Vol. 41, No. 1, Pp. 164–171.

BEKESY, G.V. (1960), *Experiments in hearing*, McGraw-Hill Book Co., New York.

BELL, A.G. (1922), 'Prehistoric telephone days', *National Geographic Magazine*, Vol. 41, Pp. 223–242.

BERGLAND, G. (1969), 'A guided tour of the fast fourier transform', *IEEE Spectrum*, Vol. 6, July, Pp. 41–52.

BEZDEL, W. and CHANDLER, H.J. (1965), 'Results of an analysis and recognition of vowels by computer using zero-crossing data', *Proc. IEE*, Vol. 112, No. 11, November, Pp. 2060–2066.

BLOMBERG, M., ELENIUS, K., LUNDSTROM, B. and NEOVIUS, L. (1987), 'Speech recognition for voice control of mobile telephone', In *European Conference on Speech Technology*, Edinburg, Scotland, U.K, September, Pp. 210–213.

BOYD, I. (1987), 'Position reoptimisation for a multipulse excited lpc coder', In LAVER, J. and JACK, M.A. (Eds.), *European Conference on Speech Technology*, CEP Consultants Ltd, Edinburgh, UK, Edinburgh, September, Pp. 37–40.

BRIESEMAN, N.P. (1984), *A new algorithm for musical pitch estimation*, Master's thesis, University of Canterbury, New Zealand.

BRIESEMAN, N.P., THORPE, C.W. and BATES, R.H.T. (1987), 'Nontactile estimation of glottal excitation characteristics of voiced speech', *IEE Proceedings A*, Vol. 134, No. 10, December, Pp. 807–813.

BRIGHAM, E.O. (1974), *The fast Fourier transform*, Prentice Hall, Englewood Cliffs, New Jersey.

BRUCKERT, E. (1984), 'A new text-to-speech product produces dynamic human-quality voice', *Speech Technology*, Vol. 2, No. 2, Jan/Feb 1984, Pp. 114–119.

BUKIET, B., RAGLAND, R.J. and DAMOULAKIS, J. (1987), 'Hardware implementation of a multi-rate real-time speech codec', In LAVER, J. and JACK, M.A. (Eds.), *European Conference on Speech Technology*, CEP Consultants Ltd, Edinburgh, UK, Edinburgh, September, Pp. 33–36.

CHANG, S., PIHL, G.E. and ESSIGMANN, M.W. (1951), 'Representattions of speech sounds and some of their statistical properties', *Proceedings of the IRE*, Vol. 39, No. 2, February, Pp. 147–153.

CLARK, T.M., KENNEDY, W.K. and BATES, R.H.T. (1990), 'Towards a real time computer word recognition system using the tms32030', In *Proceedings of the 27th National Electronics Convention*, Nelcon Incorporated, University of Auckland, Auckland, New Zealand, Sptember, Pp. 295–303.

COOLEY, J.W. and TUKEY, J.W. (1965), 'An algorithm for the machine complex Fourier series', *Math. Comput.*, Vol. 19, April, Pp. 297–301.

COOPER, F.S., LIBERMAN, A.M. and BORST, J.M. (1950), 'Preliminary studies of speech produced by a pattern playback.', *Journal of the Acoustical Society of America*, Vol. 22, P. 678.

CORCHIERE, R.E., COX, R.V. and JOHNSTON, J.D. (1982), 'Real-time speech coding', *IEEE Transactions on Communications*, Vol. COM-30, No. 4, April, Pp. 621–634. Special issue on bit rate reduction and speech interpolation.

CORNWELL, T.J. (1983), 'A method of stabilizing the clean algorithm', *Astronomy & Astrophysics*, Vol. 121, Pp. 281–285.

COX, R.V., HAGENAUER, J., SESHADRI, N. and SUNDBERG, C.E. (1988), 'A sub-band coder designed for combined source and channel coding', In *Proc. ICASSP 88*, Pp. 235–238.

CUMMISKEY, P., JAYANT, N.S. and FLANAGAN, J.L. (1973), 'Adaptive quantization in differential pcm coding of speech', *Bell System Technical Journal*, Vol. 52, Pp. 1105–1118.

DAVEY, B.L.K. (1989), *Advances in blind deconvolution*, PhD thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurchh, New Zealand.

DAVEY, B.L.K. and THORPE, C.W. (1987), 'Image and signal reconstruction by shift-and-add', In *IPENZ conference proceedings*, Institution of Professional Engineers of New Zealand, Christchurch, May.

DENES, P. and MATHEWS, M.V. (1960), 'Spoken digit recognition using time-frequency pattern matching', *Journal of the Acoustical Society of America*, Vol. 32, April, Pp. 1450–1455.

DENES, P.B. and MATHEWS, M.V. (1970), 'Laboratory computers: their capabilities and how to make them work for you.', *Proceedings of the IEEE*, Vol. 58, No. 4, April, Pp. 520–531.

DREWS, W., LAROIA, R., PANDEL, J., SCHUMACHER, A. and STOLZE, A. (1989), 'CMOS processor for template-based speech recognition system', *IEE Proceedings*, Vol. 136, Part I, No. 2, April, Pp. 155–161.

DUDLEY, H.W. (1950), 'Speech analysis and synthesis system', *Journal of the Acoustical Society of America*, Vol. 22, P. 410.

DUDLEY, H. (1955), 'Fundamentals of speech synthesis', *J. Audio Engr. Soc.*, Vol. 3, Pp. 170–185. Collected in Flanagan and Rabiner (1973).

DUDLEY, H. and TARNOCZY, T.H. (1950), 'The speaking machine of Wolfgang von Kempelen', *Journal of the Acoustical Society of America*, Vol. 22, Pp. 151–166.

DUDLEY, H., RIESZ, R.R. and WATKINS, S.A. (1939), 'A synthetic speaker', *Journal of Franklin Institute*, No. 227, Pp. 739–764. Collected in Benchmark papers in acoustics.

DURBIN, J. (1960), 'The fitting of time-series models', *Rev. Inst. Int. Statist.*, Vol. 28, No. 3, Pp. 233–243.

EWING, G.D. and TAYLOR, J.F. (1969), 'Computer recognition of speech using zero-crossing information', *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-17, No. 1, March, Pp. 37–40.

FALLSIDE, F. and WOODS, W.A. (Eds.) (1985), *Computer Speech Processing*, Prentice Hall, New Jersey.

FANT, G. (1960), *Acoustic theory of speech production*, Mouton s'Gravenhage.

FANT, G. (1973), *Speech Sounds and Features*, Current Studies in Linguistics, The MIT Press, Cambridge, Massachusetts.

FISSORE, L., LAFACE, P., MICCA, G. and PIERACCINI, R. (1989), 'Lexical access to large vocabularies for speech recognition', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-37, No. 8, August, Pp. 1197–1213.

FLANAGAN, J.L. (1972a), *Speech Analysis, Synthesis and Perception*, Kommunikation und Kybernetic in Einzeldarstellungen, Springer-Verlag, Berlin, 2nd ed.

FLANAGAN, J.L. (1972b), 'Voices of men and machines', *The Journal of the Acoustical Society of America*, Vol. 51, Pp. 1375–1387.

FLANAGAN, J.L. (1972c), 'The synthesis of speech', *Scientific American*, Vol. 226, No. 2, February, Pp. 48–56.

FLANAGAN, J.L. and LANDGRAF, L.L. (1968), 'Self-oscillating source for vocal-tract synthesis', *IEEE Trans. of Audio Engineers*, Vol. AU-16, Pp. 57–64.

FLANAGAN, J.L. and RABINER, L.R. (Eds.) (1973), *Speech synthesis*, Benchmark papers in acoustics, Dowden, Hutchinson and Ross, Inc., Stroudsburg, Pennsylvania.

FLANAGAN, J.L., RABINER, L.R., SCHAFER, R.W. and DENMAN, J.D. (1972), 'Wiring telephone apparatus from computer-generated speech', *BSTJ*, Vol. 51, No. 2, February, Pp. 391–7.

FLANAGAN, J.L., SCHROEDER, M.R., ATAL, B.S., CROCHIERE, R.E., JAYANT, N.S. and TRIBOLET, J.M. (1979), 'Speech coding', *IEEE Transactions on Communications*, Vol. COM-27, No. 4, April, Pp. 710–737.

FLETCHER, H. (1940), 'Auditory patterns', *Journal of the Acoustical Society of America*, Vol. 12, Pp. 47–65.

FORNEY, Jr., G.D. (1973), 'The Viterbi algorithm', *Proceedings of the IEEE*, Vol. 61, No. 3, March, Pp. 268–278.

FUKUI, A. and SHIBAGAKI, K. (1987), 'Speech quality improvement of a mutil-pulse speech codec with pitch prediction on a single chip signal processor', In LAVER, J. and JACK, M.A. (Eds.), *European Conference on Speech Technology*, CEP Consultants Ltd, Edinburgh, UK, Edinburgh, September, Pp. 41–44.

FULTZ, K.E. and PENICK, D.B. (1965), 'T1 carrier system', *Bell System Tech. Jour.*, Vol. 44, September, Pp. 1405–1451.

FURUI, S. (1981), 'Cepstral analysis technique for automatic speaker verification', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 2, April, Pp. 254–272.

FURUI, S. (1988), 'A VQ-based preprocessor using cepstral dynamic features for speaker-independent large vocabulary word recognition', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-36, No. 7, July, Pp. 980–987.

GERSHO, A. (1982), 'On the structure of vector quantizers', *IEEE Trans. Inform Theory*, Vol. IT-28, March, Pp. 157–166. Special issue on quantization.

GERSHO, A. and RAMAMURTHI, B. (1982), 'Image coding using vector quantization', In *International conference on speech and signal processing*, Pp. Vol 1, 428--431.

GOLD, B. and RABINER, L.R. (1968), 'Analysis of digital and analog formant synthesisers', *IEEE Trans of Audio Engineers*, Vol. AU-16, Pp. 81–94.

GOLD, B. and RABINER, L.R. (1969), 'Parallel processing techniques for estimating pitch periods of speech in the time domain', *Journal of Acoustical Society of America*, Vol. 46, Pp. 442–448.

GOLD, B. and RADER, C.M. (1967), 'The channel vocoder', *IEEE Transaction of Audio Engineers*, Vol. AU-15, Pp. 148–161.

GRAF, H.P., JACKEL, L.D. and HUBBARD, W.E. (1988), 'VLSI implementation of a neural network model', *IEEE Transactions in Computers*, Vol. 21, No. 3, March, Pp. 41–51. Special issue on artificial neural systems.

GRANT, P.M. (1991), 'Speech recognition techniques', *Electronics & Communication Engineering Journal*, Vol. 3, No. 1, Pp. 37–48.

GRAY, R.M. (1984), 'Vector quantization', *IEEE Acoustics, Speech and Signal Processing Society Magazine*, April, Pp. 4–28.

GRAY, A.H. and MARKEL, J.D. (1979), 'Linear prediction analysis of speech signals', In Digital Signal Processing Committee, IEEE ASSP Society (Ed.), *Programs for digital signal processing*, IEEE Press, Chap. 4.

GREENWOOD, D.D. (1961), 'Critical bandwidths and frequency co-ordinates of the basilar membrane', *Journal of the Acoustical Society of America*, Vol. 33, Pp. 1344–1356.

GRENANDER, U. and SZEGö, G. (1958), *Toeplitz forms and their applications*, University of California Press, Berkeley, California.

GRUENZ, O. and SCOTT, L.O. (1949), 'Extraction and portrayal of pitch of speech sounds', *Journal of the Acoustical Society of America*, Vol. 21, Pp. 487–495.

GULAMHUSEIN, M.N. (1973), *Short-time Walsh analysis-synthesis with applications to digital speech processing*, PhD thesis, Cambridge University.

GUNAWARDANA, R. (1987), 'Low cost cmos voice synthesis processors: a device family for applications in consumer to telecommunications fields', In *Official Proceedings of International Speech Tech'87*, Voice Input/Output Applications Show and Conference, Media Dimensions, New York, U.S.A., 26-28 May, Pp. 67–70.

HALLE, M. and STEVENS, K.N. (1959), 'Analysis by synthesis', In WATHEN-DUNN, W. and WOODS, L.E. (Eds.), *Proceedings of the seminar on speech compression and processing*, AFCRC-TR-59-198. Vol II, paper D7.

HARRIS, F. (1978), 'On the use of windows for harmonic analysis with discrete Fourier transform', *Proc. IEEE*, Vol. 66, No. 1, January.

HAWKING, S.W. (1987), *A brief history of time*, Bantam Books, Great Britain.

HAYKIN, S. (1983), *Communication systems*, John Wiley & Sons, Brisbane.

HAYKIN, S. (1989), *An introduction to analog and digital communications*, John Wiley & Sons, Brisbane.

HERMANSKY, H. (1987), 'Automatic speech recognition and human auditory perception', In *European Conference on Speech Technology*, Edinburg, Scotland, U.K., September, Pp. 79–82.

HESS, W.H. (1983), *Pitch determination of speech signals - algorithms and devices*, Springer-Verlag.

HILLS, D.A. and SHIPLEY, E.D. (1985), *Sound*, Technical and Liaison Bulletin Series 5, Southern Industrial Development Division, Department of Scientific and Industrial Research, New Zealand.

HöGBOM, J.A. (1974), 'Aperture synthesis with a non-regular distribution of interferometer baselines', *Astronomical Astrophysics Supplement*, Vol. 15, Pp. 417–426.

HOUTSMA, A.J.M. and GOLDSTEIN, J.L. (1972), 'The central origin of the pitch of complex tones: Evidence from musical interval recognition', *Journal of the Acoustical Society of America*, Vol. 51, No. 2 (Pt. 2), February, Pp. 520–529.

HUANG, X.D. and JACK, M.A. (1988), 'Hidden Markov modelling of speech based on a semicontinuous model', *Electronics Letters*, Vol. 24, No. 1, 7th January, Pp. 6–7.

HUFFMAN, D.A. (1973), 'A method for the construction of minimum-redundancy codes', *Proc. IRE*, Vol. 40, September, Pp. 1098–1101.

IEE (1985), 'Electrical & electronics abstract, cumulative subject index', The Institution of Electrical Engineers.

IEEE (1969), 'Recommended practice for speech quality measurements', *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-17, No. 3, September, Pp. 225–246.

ITAKURA, F. (1975), 'Minimum prediction residual principle applied to speech recognition', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-23, No. 1, Feb, Pp. 67–72.

JAYANT, N.S. (1973), 'Adaptive quantization with a one-word memory', *Bell System Tech. Journal*, Pp. 1119–1144.

JAYANT, N.S. (1986), 'Coding speech at low bit rates', *IEEE Spectrum*, August, Pp. 58–63.

JAYANT, N.S. (1990), 'High-quality coding of telephone speech and wideband audio', *IEEE Communications Magazine*, January, Pp. 10–16.

JAYANT, N.S. and NOLL, P. (1984), *Digital coding of waveforms - Principles and applications to speech and video*, Prentice-Hall Inc., Englewood Cliffs, N.J.

JEFFRESS, L.A. (1970), 'Masking', In TOBIAS, J.V. (Ed.), *Foundations of modern auditory theory*, Academic Press.

JELINIK, F. (1976), 'Continuous speech recognition by statistical methods', *Proceedings of the IEEE*, Vol. 64, No. 4, April, Pp. 532–556.

JUANG, B.H. (1985), 'Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains', *AT & T Technical Journal*, Vol. 64, No. 6, July-August, Pp. 1235–1249.

JUANG, B.H., RABINER, L.R., LEVINSON, S.E. and SONDHI, M.M. (1985), 'Recent developments in the application of hidden Markov models to speaker-independent isolated word recognition', In *International conference on speech and signal processing*, IEEE, ASSP Society, Tampa, Florida, March, Pp. 1.3.1–1.3.4.

KANEKO, H. (1970), 'A unified formulation of segment companding laws and synthesis of codecs and digital companders', *Bell System Technical Journal*, Vol. 49, September, Pp. 1555–1588.

KAY, S.M. and MARPLE, S.L. (1981), 'Spectrum analysis—A modern perspective', *Proceedings of the IEEE*, Vol. 69, No. 11, November, Pp. 1380–1419.

KEMPELEN, W.V. (1791), *Mechanismus der menschlichen Sprache nebst der Beschreibung seiner sprechenden Maschine*, ???

KIMURA, D. (1961), 'Cerebral dominance and the perception of verbal stimuli', In HAWLEY, M.E. (Ed.), *Speech intelligiblity and speaker recognition*, Dowden, Hutchinson and Ross, Inc, Pennsylvania, Pp. 108–113.

KITAWAKI, N. and NAGABUCHI, H. (1988), 'Quality assessment of speech coding and speech synthesis systems', *IEEE Communications Magazine*, October, Pp. 36–44.

KITAWAKI, N., HONDA, M. and ITOH, K. (1984), 'Speech-quality assessment methods for speech-coding systems', *IEEE Communications Magazine*, Vol. 22, No. 10, October, Pp. 26–33.

KOENIG, W., DUNN, H.K. and LACY, L.Y. (1946), 'The sound spectograph', *J. Acoust. Soc. Amer.*, Vol. 18, Pp. 19–49.

KOHONEN, T. (1988), 'The neural phonetic typewriter', *IEEE Transactions in Computer*, Vol. 21, No. 3, March, Pp. 19–49. Special issue on Artificial Neural Systems.

KOHONEN, T., MAKISARA, K. and SARAMAKI, T. (1984), 'Phonotopic maps - Insightful representations of phonological features for speech recognition', In *Proceedings of the seventh internation conference on pattern recognition, Vol. 1*, Canadian Image Processing and Pattern Recognition Society and International Association for Pattern Recognition, North-Holland/IEEE Computer Society Press, Montreal, Canada, July 30 - August 2, Pp. 182–185.

KONDOZ, A.M. and EVANS, B.G. (1988), 'Celp base-band coder for high quality speech coding at 9.6 to 2.4 kbps', In *ICASSP 88 International Conference on Acoustics, Speech, and Signal Processing, Volume 1: Speech Processing*, IEEE-ASSP Society, Tokyo, Japan, Pp. 159–161.

KREYSZIG, E. (1983), *Advanced Engineering Mathematics*, John Wiley & Sons, 5th edition ed.

KROON, P., DEPRETTERE, E.F. and SLUYTER, R.J. (1986), 'Regular-pulse excitation - a novel approach to effective and efficient multipulse coding of speech', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-34, No. 5, October, Pp. 1054–1063.

LAFUENTE, L.M. (1983), 'Adaptive differential pulse code modulation coder for low bit rate transmission of speech signals', *Electrical Communications*, Vol. 24, Pp. 225–229.

LAMB, M.R. and BATES, R.H.T. (1978), 'Computerized aural training: an interactive system designed to help both student and teachers', *Journal of Computer-Based Instruction*, Vol. 5, No. 1 and 2, Aug and Nov, Pp. 30–37.

LAVER, J., HILLER, S. and MACKENZIE, J. (1984), 'Acoustic analysis of vocal fold pathology', *Proc. Inst. Acoust.*, Vol. 6, Pp. 425–430.

LEE, C.H. (1988), 'On robust linear prediction of speech', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-36, No. 5, May, Pp. 642–650.

LEE, K., HON, H. and REDDY, R. (1990), 'An overview of the SPHINX speech recognition system', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. 38, No. 1, January, Pp. 35–45.

LEFEVRE, J. and PASSIEN, O. (1985), 'Efficient algorithms for obtaining multipulse excitation for lpc coders', In *Proc. Int. Conf. Acoust., Speech and Signal Processing*, Tampa, Florida, Pp. 957–960.

LEVINSON, N. (1947), 'The Wiener rms (root mean square) error criterion in filter design and prediction', *J. Math. Phys.*, Vol. 25, No. 4, Pp. 261–278.

LEVINSON, S.E. and ROE, D.B. (1990), 'A perspective on speech recognition', *IEEE Communications Society Magazine*, Vol. 28, No. 1, January, Pp. 28–34. Special issue on speech processing and applications.

LEVINSON, S.E., RABINER, L.R. and SONDHI, M.M. (1983a), 'Speaker-independent digit recognition using hidden Markov models', *Proc. ICASSP 1983*, Pp. 1049–1052.

LEVINSON, S.E., RABINER, L.R. and SONDHI, M.M. (1983b), 'An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition', *Bell System Technical Journal*, Vol. 62, No. 4, April, Pp. 1035–1074.

LIBERMAN, A.M. and MATTINGLY, I.G. (1985), 'The motor theory of speech revised', *Cognition*, Vol. 21, Pp. 1–36.

LIBERMAN, A.M., COOPER, F.S., SHANKWEILER, D.P. and STUDDERT-KENNEDY, M. (1967), 'Perception of the speech code', *Psychological Review*, Vol. 74, Pp. 431–461.

LICKLIDER, J.C.R. (1960), 'Effects of differentiation, integration and infinite peak clipping upon the intelligibility of speech', *Journal of the Acoustical Society of America*, Vol. 20, P. 42.

LIEBERMAN, P. and BLUMSTEIN, S.E. (1988), *Speech physiology, speech perception, and acoustic phonetics*, Cambridge studies in speech science and communication, Cambridge University Press, Cambridge, U.K.

LIFSCHITZ, S. (1933), 'Two integral laws of sound perception relating loudness and apparent duration of sound impulses', *Journal of Acoustical Society of America*, Vol. 7, Pp. 213–219.

LIM, C.A., ELDER, A.G., CLARK, T.M. and BATES, R.H.T. (1990), 'Software implementation of hidden markov model for recognition of isolated digits uttered by New Zealand speaker', In *Proceedings of the 27th National Electronics Convention*, NELCON INCORPORATED, University of Auckland, Auckland, New Zealand, Spetempber, Pp. 287–294.

LINDE, Y., BUZO, A. and GRAY, R.M. (1980), 'An algorithm for vector quantiser design', *IEEE Transactions on Communications*, Vol. COM-28, No. 1, January, Pp. 84–95.

LINGGARD, R. (1985), *Electronic synthesis of speech*, Cambridge University Press, Cambridge, U.K.

LIPPMANN, R.P. (1987), 'An introduction to computing with neural nets', *IEEE Acoustics, Speech and Signal Processing Society Magazine*, Vol. 4, No. 2, April, Pp. 4–22.

LIPPMANN, R.P. (1988), 'Neural nets for computing', In *International conference on speech and signal processing*, IEEE, ASSP Society, New York City, April, Pp. 1–6.

LLOYD, S.P. (1982), 'Least squares quantization in pcm', *IEEE Trans. Inform. Theory*, Vol. IT-28, No. 2, March, Pp. 129–137.

LUCK, J.E. (1969), 'Automatic speaker verification using cepstral measurements', *Journal of the Acoustical Society of America*, Vol. 46, No. 4, Pp. 1026–1032.

MAKHOUL, J. (1973), 'Spectral analysis of speech by linear prediction', *IEEE Trans. Audio Electroacoust.*, Vol. AU-21, June, Pp. 140–148.

MAKHOUL, J. (1975), 'Linear prediction: A tutorial review', *Proceedings of the IEEE*, Vol. 63, No. 4, April, Pp. 561–580.

MAKHOUL, J., ROUCOS, S. and GISH, H. (1985), 'Vector quantisation in speech coding', *Proceedings of the IEEE*, Vol. 73, No. 11, November, Pp. 1551–1558.

MARKEL, J.D. and A. H. GRAY, J. (1973), 'On autocorrelation equations as applied to speech analysis', *Institute of Electrical and Electronic Engineers, Trans*, Vol. AU-21, Pp. 69–79.

MARKEL, J.D. and A. H. GRAY, J. (1976), *Linear prediction of speech*, Springer-Verlag, Berlin.

MAYER, A.M. (1894), 'Researches in acoustics', *Lond. Edinb. Dubl. Phil. Mag.*, Vol. 37, No. ser. 5, Pp. 259–288.

McADAM, D.W. and WHITAKER, H.A. (1971), 'Language production: Electroencephalographic localization in the normal human brain', *Science*, Vol. 172, Pp. 499–502.

MCCANDLES, S.S. (1974), 'An algorithm for automatic formant extraction using linear predictive spectra', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-22, Pp. 135–141.

MCINNES, F.R., JACK, M.A. and LAVER, J. (1989), 'Template adaptation in an isolated word-recognition system', *IEE Proceedings-I:Communications, Speech and Vision*, Vol. 136, No. 2, April, Pp. 119–126.

MCLAUGHLIN, P.T. (1981), 'A single chip speech synthesis system', In *24th Midwest Symposium on Circuits and Systems*, Western Periodicals, North Hollywood, CA, U.S.A., 29-30 June, Pp. 751–756.

MILLER, R.L. (1959), 'Nature of the vocal cord wave', *Journal of the Acoustical Society of America*, Vol. 31, No. 6, June, Pp. 667–677.

MINARD, R.A., ROBINSON, B.S. and BATES, R.H.T. (1985), 'Full-wave computed tomography part 3: coherent shift-and-add imaging', *IEE Proc. Part A*, Vol. 132A, Pp. 50–58.

MOORE, R.K., RUSSEL, M.J. and TOMLINSON, M.J. (1982), 'Locally constrained dynamic programming in automatic speech recognition', In *International conference on speech and signal processing*, Pp. 1270–1273.

MUNSON, W.A. and MONTGOMERY, H.C. (1950), 'Speech analyzer and synthesizer', *Journal of the Acoustical Society of America*, Vol. 22, P. 678.

MURAKAMI, T., ASAI, K. and YAMAZAKI, E. (1982), 'Vector quantizer of video signals', *Electronic Letters*, Vol. 7, November, Pp. 1005–1006.

NAKANO, T., HONDA, T. and OGAWA, Y. (1970), *Constitution of speech synthesizer for deaf and dumb persons*, Research Report of The Faculty of Engineering 25, Pp 149-57, Meiji University, Japan. In Japanese. Abstract appeared in Electrical and Electronics Abstract Journal.

NEOH, K.H. (1973), *Synthetic speech for the handicapped*, Master's thesis, School of Engineering, University of Canterbury, February.

NOLL, A.M. (1964), 'Short-time spectrum and "cepstrum" techniques for vowel pitch detection', *Journal of the Acoustical Society of America*, Vol. 36, Pp. 296–302.

O'CONNOR, J. (1990), 'Talking adverts about to startle unwary', *Christchurch Press*, January 31, P. 21.

OIZUMI, J. and KUBO, E. (1954), 'Synthesizing speech', *Journal of Acoust. Soc. Japan*, Vol. 10, September, Pp. 155–158. In Japanese with English abstract.

O'MALLEY, M.H. and KLOKER, D. (1971), 'Speech synthesis and editing system for teaching phonetics', In *81st Meetings of the Acoustical Society of America*, Acoustical Society of America, Washington, D.C., 20-23 April 1971, P. 31.

OPPENHEIM, A.V. (1970), 'Speech spectograms using the fast Fourier transform', *IEEE Spectrum*, Vol. 7, August, Pp. 57–62.

OPPENHEIM, A.V. and SCHAFER, R.W. (1975), *Digital signal processing*, Prentice-Hall Inc., New Jersey.

PAGET, R. (1930), *Human speech: some obervations, experiments and conclusions as to the nature, origin, purpose and possible improvement of human speech*, Harcourt, New York.

PAUL, D.B. (1985), 'Training of HMM recognizers by simulated annealing', In *International conference on speech and signal processing*, IEEE, ASSP Society, Tampa, Florida, March, Pp. 13–16.

PICONE, J. (1990), 'Continuous speech recognition using hidden Markov models', *IEEE ASSP Magazine*, Vol. 7, No. 3, July, Pp. 26–41.

POLLACK, I. (1952), 'The information of elementary audio displays', *Journal of the Acoustical Society of America*, Vol. 24, Pp. 745–749.

PORITZ, A.B. (1988), 'Hidden Markov models: A guided tour', In *International conference on speech and signal processing*, IEEE, ASSP Society, New York city, April, Pp. 7–13.

PRICE, J.P.J., FISHER, W., BERNSTEIN, J. and PALLETT, D. (1988), 'The DARPA 1000-word resource management database for continuous speech recognition', In *International conference on speech and signal processing*, New York City, April, Pp. 651–654.

PTACEK, M. (1972), 'The tesla transistorized speech synthesizer', *Sdelovaci Tech.*, Vol. 19, No. 12, Dec, Pp. 394–6. In Czech. Abstract appeared in Electrical & Electronics Abstract Journal, 1972.

RABINER, L.R. (1968), 'Digital-formant synthesiser for speech synthesis studies', *Journal of the Acoustical Society of America*, Vol. 43, Pp. 822–828.

RABINER, L.R. (1977), 'On the use of autocorrelation analysis for pitch detection', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-25, Pp. 24–33.

RABINER, L.R. (1989), 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE*, Vol. 77, No. 2, February, Pp. 257–286.

RABINER, L.R. and JUANG, B.H. (1986), 'An introduction to hidden Markov models', *IEEE Acoustics, Speech and Signal Processing Society Magazine*, Vol. 3, No. 1, January, Pp. 4–16.

RABINER, L.R. and LEVINSON, S.E. (1981), 'Isolated and connected word recognition- theory and selected applications', *COM*, Vol. COM-29, No. 5, May, Pp. 621–659.

RABINER, L.R. and LEVINSON, S.E. (1985), 'A speaker-independent, syntax directed, connected word recognition system based on hidden Markov models and level building', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-33, No. 3, June, Pp. 561–573.

RABINER, L.R. and SCHAFER, R.W. (1978), *Digital signal processing of speech signals*, Prentice-Hall, Englewood Cliffs, New Jersey 07632, USA.

RABINER, L.R. and WILPON, J.G. (1987), 'Some performance benchmarks for isolated word speech recognition systems', *Computer Speech and Language*, Vol. 2, Pp. 343–357.

RABINER, R., JACKSON, B., SCHAFER, R.W. and COKER, C.H. (1971), 'Digital hardware fro speech synthesis', In *Proceedings of the 7th International Congress on Acoustics*, Budapest, Hungary, 18-26 August, Pp. 157–60.

RABINER, L.R., ROSENBERG, A.E. and LEVINSON, S.E. (1978), 'Considerations in dynamic time warping algorithms for discrete word recognition', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-26, No. 6, December, Pp. 575–582.

RABINER, L.R., LEVINSON, S.E., ROSENBERG, A.E. and WILPON, J.G. (1979), 'Speaker-independent recognition of isolated words using clustering techniques', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-27, August, Pp. 336–349.

RABINER, L.R., LEVINSON, S.E. and SONDHI, M.M. (1983), 'On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition', *Bell Systems Technical Journal*, Vol. 62, No. 4, April, Pp. 1075–1105.

RABINER, L.R., WILPON, J.G. and JUANG, B.H. (1986), 'A segmental k-means training procedure for connected word recognition', *AT & T Technical Journal*, Vol. 65, No. 3, May/June, Pp. 21–31.

RABINER, L.R., WILPON, J.G. and SOONG, F.K. (1989), 'High performance connected digit recognition using hidden Markov models', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. 37, No. 8, August, Pp. 1214–1225.

RAMAMOORTHY, V. (1985), 'A novel speech coder for medium and high bit rate applications using modulo-PCM principles', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-33, No. 4, April, Pp. 356–368.

RAMSTAD, T.A. (1982), 'Sub-band coder with a simple adaptive bit allocation algorithm', In *Proc. ICASSP 82*, Pp. 203–207.

ROSENBERG, A.E. and ITAKURA, F. (1976), 'Evaluation of an automatic word recognition system over dialled-up telephone lines', *Journal of the Acoustical Society of America*, Vol. 60, No. suppl 1, P. S12.

SAKOE, H. and CHIBA, S. (1978), 'Dynamic programming algorithm optimization for spoken word recognition', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-26, No. 1, February, Pp. 43–49.

SCHROEDER, M.R. (1968a), 'Period histogram and product spectrum: new methods for fundamental frequency measurement', *Journal of Acoustical Society of America*, Vol. 43, Pp. 829–834.

SCHROEDER, M.R. (1968b), 'Reference signal for signal quality studies', *Journal of the Acoustical Society of America*, Vol. 44, No. 6, Pp. 1735–1736.

SCHROEDER, M.R. (1969), 'Computers in acoustics: symbiosis of an old science and a new tool', *Journal of the Acoustical Society of America*, Vol. 45, May, Pp. 1077–1088.

SCHROEDER, M.R. (1984), 'Linear prediction, entropy and signal analysis', *IEEE Acoustics, Speech and Signal Processing Society Magazine*, Vol. 1, No. 3, July, Pp. 3–11.

SCHROEDER, M.R. (1985), 'Linear predictive coding of speech: Review and current directions', *IEEE Communications Magazine*, Vol. 23, No. 8, August, Pp. 54–61.

SHANNON, C.E. (1948), 'A mathematical theory of communication', *Bell Syst. Tech. J.*, Vol. 27, Pp. 379–423, 623–656.

SHANNON, C.E. (1949), 'Communication in the presence of noise', *Proceedings of the Institute of Radio Engineers*, Vol. 37, January, Pp. 10–21.

SHARF, B. (1970), 'Critical bands', In TOBIAS, J.V. (Ed.), *Foundations of modern auditory theory*, Academic Press, New York.

SINGHAL, S. and ATAL, B.S. (1989), 'Amplitude optimization and pitch prediction in multipulse coders', *assp*, Vol. 37, No. 3, March, Pp. 317–327.

SMITH, B. (1957), 'Instantaneous companding of quantized signals', *Bell System Technical Journal*, Vol. 36, May, Pp. 653–709.

SOONG, F.K., ROSENBERG, A.E., JUANG, B.H. and RABINER, L.R. (1987), 'A vector quantization approach to speaker recognition', *AT & T Technical Journal*, Vol. 66, No. 2, March/April, Pp. 14–26.

STEVENS, S.S. and DAVIS, H. (1938), *Hearing*, New York: John Wiley and Sons.

STEWART, J.Q. (1922), 'An electrical analysis of the vocal organs', *Nature*, Vol. 110, Pp. p311–12.

STREMLER, F.G. (1982), *Introduction to communication systems*, Addison-Wesley Publishing Company, Sydney, second ed.

SUTHERLAND, A.M., JACK, M.A. and LAVER, J. (1988), 'Improved pitch detection algorithm employing temporal structure investigation of the speech waveform', *IEE Proceedings F*, Vol. 135, No. 2, April, Pp. 169–174.

SYKES, J.B. (Ed.) (1976), *The concise Oxford dictionary of current English*, Oxford University Press, Oxford, UK, 6th edition ed.

TAN, W.M. (1990), *Investigation and evaluation of a new low bit rate speech coding scheme*, Technical Report, Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, October.

TEXAS INSTRUMENTS (1983), *TMS32010 User's Guide - Digital Signal Processor Product*, Texas Instruments Incorporated.

THORPE, C.W. (1990), *Analysis of speech and other sounds*, PhD thesis, Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, November.

TRANCOSO, I.M. and TRIBOLET, J.M. (1989), 'Harmonic postprocessing of speech synthesised by stochastic coders', *IEE Proceedings-I:Communications, Speech and Vision*, Vol. 136, No. 2, April, Pp. 141–145.

TRUPP, R.D. (1970), 'Computer-controlled message synthesis', *Bell Lab. Record*, June/July, Pp. 175–180.

TUCKER, W. and BATES, R. (1978), 'A pitch estimation alogrithm for speech and music', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. ASSP-26, Pp. 597–604.

TUCKER, W., BATES, R.H.T., FRYKBERG, S.D., HOWARTH, R.J., KENNEDY, W.K., LAMB, M.R. and VAUGHAN, R.G. (1977), 'An interactive aid for musicians', *International Journal Man-Machine Studies*, Vol. 9, Pp. 635–651.

TURNER, S.G. (1986a), *Real-time speech analysis for use with impaired speech aids*, Master's thesis, Electrical and Electronic Engineering, University of Canterbury, NZ, March.

TURNER, S.G. (1986b), *Real-time speech analysis for use with impaired speech aids*, Master's thesis, Electrical and Electronic Engineering, University of Canterbury, NZ, March.

WAIBEL, A., HANAZAWA, T., HINTON, G., SHIKANO, K. and LANG, K.J. (1989), 'Phoneme recognition using time-delay neural networks', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. 37, No. 3, March, Pp. 328–339.

WALLICH, P. (1987), 'Putting speech recognisers to work', *IEEE Spectrum*, April, Pp. 55–57.

WATSON, C.I., CLARK, T.M., ELDER, A.G. and THORPE, C.W. (1988), 'Multifarious real-time speech processing applications', In *Proc. National Electronics Conference*, Proc. NELCON, (New Zealand National Electronics Conference), Christchurch, September, Pp. 65–70.

WEGEL, R.L. and LANE, C.E. (1924), 'The auditory masking of one pure tone by another and its probable relation to the dynamics of the inner ear', *Phys. Rev.*, Vol. 23, No. ser. 2, Pp. 266–285.

WHEATSTONE, C. (1879), 'Scientific papers of Sir Charles Wheatstone'. Royal Society of London, London.

WIENER, F.M. and ROSS, D.A. (1946), 'The pressure distributon in the auditory canal in a progressive sound field', *Journal of the Acoutical Society of America*, Vol. 18, Pp. 401–408.

WINCKEL, F. (1951), 'Eletric-synthetic production of speech sounds', *Z. f. Phonetik*, Vol. 5, May-Aug, Pp. 257–263. in German.

WITTEN, I.H. (1982), *Principles of computer speech*, Academic Press, London.

WONG, D.Y. and MARKEL, J.D. (1977), 'An intelligiblity evaluation of several linear prediction vocoder modifications', In *1977 IEEE Internation Conference on Acoustics, Speech & Signal Processing*, IEEE, Hartford, Connecticut, May 9-11, Pp. 208–211.

WONG, D.Y., JUANG, B. and GRAY, JR., A.H. (1982), 'An 800 bit/s vector quantisation LPC vocoder', *IEEE Transactions in Acoustics, Speech and Signal Processing*, Vol. 30, No. 5, October, Pp. 770–780.

WOOD, L.C. and PEARCE, D.J.B. (1989), 'Excitation synchronous formant analysis', *IEE Proceedings-I:Communications, Speech and Vision*, Vol. 136, No. 2, April, Pp. 110–118.

ZWICKER, E. (1961), 'Subdivision of the audible frequency range into critical bands', *Journal of the Acoustical Society of America*, Vol. 33, P. 248.

ZWISLOCKI, J. (1965), 'Analysis of some auditory characteristics', In LUCE, R.D., BUSH, R.R. and GALANDER, E. (Eds.), *Handbook of mathematical psychology, Vol 3*, Wiley, New York.